

AD-A172 652

CSS/EMW/SOF (COMBAT SERVICE SUPPORT/ENGINEERING AND
MINE WARFARE/SPECIAL... (U) MCLEAN RESEARCH CENTER INC VA
G N ROMSTEDT SEP 86 DAAK70-84-D-0052

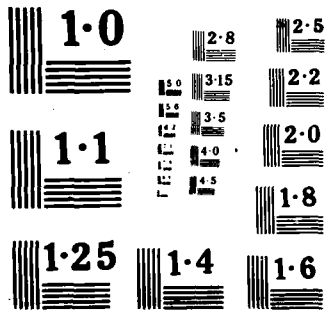
1/4

UNCLASSIFIED

F/G 9/2

NL





2

CSS/EMW/SOF

MISSION AREA MATERIEL PLAN (MAMP)

SOFTWARE

AD-A172 652

DTIC
ELECTE
OCT 06 1986
S D

DTIC FILE COPY

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

McLEAN RESEARCH CENTER, INC.
1483 Chain Bridge Road, Suite 205
McLean, Virginia 22101

394691 PTM

86 10 3 006

DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

2

DTIC
ELECTE
OCT 06 1986
S D D

CSS/EMW/SOF
MISSION AREA MATERIEL PLAN (MAMP)
SOFTWARE

September, 1986

Prepared for
US Army Belvoir Research, Development and Engineering Center
Fort Belvoir, Virginia 22060-5606

by
G. Neil Romstedt

McLean Research Center, Inc.
1483 Chain Bridge Road, Suite 205
McLean, Virginia 22101 (703) 734-1410

This report documents the research performed under Contract Number DAAK70-84-D-0052, Task Order 0016. The views, opinions, and/or findings contained in this report are those of the authors and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A172 652

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) MRC (DAAK70-84-D-0052, Task 016)		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION McLean Research Center, Inc.	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION DCASMA	
6c. ADDRESS (City, State, and ZIP Code) 1483 Chain Bridge Road, Suite 205 McLean, Virginia 22101		7b. ADDRESS (City, State, and ZIP Code) 300 East Joppa Road, Suite 200 Towson, Maryland 21204	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION BRDEC	8b. OFFICE SYMBOL (If applicable) STRBE-HP	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER DAAK70-84-D-0052, Task 0016	
8c. ADDRESS (City, State, and ZIP Code) Fort Belvoir, Virginia 22060-5606		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO.	PROJECT NO.
		TASK NO.	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) (U) CSS/EMW/SOF Mission Area Materiel Plan (MAMP) Software			
12. PERSONAL AUTHOR(S) G.N. Romstedt			
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM FEB 86 TO SEP 86	14. DATE OF REPORT (Year, Month, Day) 1986 September	15. PAGE COUNT 316
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	MAMP Program Planning Budget Planning	
		Database Management System Prioritization	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>This report documents the status of the Combat Service Support (CSS)/Engineering and Mine Warfare (EMW)/Special Operations Forces (SOF) Mission Area Material Plan (MAMP) Software. This software is used for program planning and resource allocation AMC RDT&E initiatives. It also presents an analysis of some of the "system" aspects of the automated MAMP as it is instituted throughout AMC, and of the program prioritization methods used.</p>			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL K. J. DEAN		22b. TELEPHONE (Include Area Code) (703) 664-4458	22c. OFFICE SYMBOL STRBE-HP

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

STUDY GIST

Management Science Support for Program/Budget Planning and Prioritization Methodology

Principal Findings

1. The greatest potential for the MAMP is the close cooperation it engenders between the AMC materiel developer and the TRADOC combat developer in establishing goals and recommending programs to DA. In many cases, there has been insufficient communication between these parties. The MAMP provides a formal mechanism and a visible product of that communication which is less demanding than the preparation of requirements documents. As a planning tool, the MAMP excels in visualizing program structure and resource requirements for the future.
2. The MAMP software is still under development. The exact informational requirements of MAMP users have not yet been fully ascertained, much less satisfied. The manager is unlikely to know what the best presentation of information is until he sees a sample report. Thus, the work done at BRDEC can be viewed as the development of prototype reports which can be critiqued, improved, and adopted if they are successful.
3. The data contained in the MAMP database requires a lot of work to assemble and maintain. The quality control functions require a significant degree of knowledgeable high level attention.

Main Assumptions

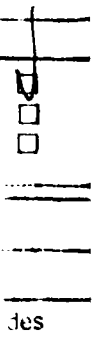
1. The BRDEC automated MAMP system can be made to be responsive to BRDEC and CSS/EMW/SOF, independently of software changes that may occur in the AMC MAMP. This specialized software will, however, have to conform with standards set by the AMC system.
2. Any specialized new reports, if they prove to be responsive to informational needs, can become prototypes for additions to the AMC system.

Principal Limitations

1. The MAMP software available at BRDEC is generally unavailable elsewhere. This is because of the somewhat technical nature of the computer operating system and the complexity of the software. It will require integration into the AMC MAMP system for non-technical users to receive much benefit.
2. The MAMP program prioritization algorithm is the responsibility of TRADOC. This study analyzed the perceived deficiencies in the current algorithm, but it was felt that the technique could not be unilaterally altered.



Dist	Approved/	or
	Special	
A-1	22	20



Scope of the Effort

This effort was directed toward improving the automated MAMP system in place at BRDEC and applied to the management of the CSS, EMW, and SOF mission areas.

Objective

The objectives of this effort were to provide management science support to test, analyze, and apply the revised AMC MAMP software; to identify and implement extensions to the MAMP that address BRDEC unique requirements; and to further develop alternative methods to integrate the program prioritization and funding allocation processes.

Basic Approach

(1) Functional and comparative analyses of the original BRDEC and revised AMC MAMP software were performed to identify all substantive differences in their content and performance. The differences were such that it was determined that the BRDEC MAMP would be made to conform structurally with the AMC MAMP.

(2) Extensions to the MAMP software were added so that the MAMP would be applicable to the BRDEC unique requirements that span across several TRADOC Mission Areas.

(3) Other program prioritization techniques, specifically those that integrate the MAMP into the DA LRRDAP process, were reviewed and implemented to assist in the assessment of reprogramming recommendations.

Reason for Performing the Study

To provide the necessary support to the Mission Area Manager for the CSS, EMW, and SOF mission areas, in the continued development and production of the Mission Area Materiel Plan.

Sponsor

US Army Belvoir Research, Development and Engineering Center
Fort Belvoir, Virginia 22060-5606

Principal Investigator

G. Neil Romstedt, McLean Research Center, Inc.

Comments and Questions

McLean Research Center, Inc.
1483 Chain Bridge Road, Suite 205
McLean, Virginia 22101
(703) 734-1410

DLSIE Accession Number of Final Report

N/A (Applied for 9/30/86)

FOREWORD

This technical report is submitted to the US Army Belvoir Research, Development and Engineering Center, Fort Belvoir, Virginia by the McLean Research Center, Inc. (MRC), 1483 Chain Bridge Road, Suite 205, McLean, Virginia. This report documents the research performed under Contract Number DAAK70-84-D-0052, Task Order 0016 entitled "Management Science Support for Program/ Budget Planning and Prioritization Methodology". The major portion of the research effort is presented in the four appendices which document the status of the CSS/EMW/SOF MAMP software as of August 1986. They are intended for use as a Reference Manual for system operators.

TABLE OF CONTENTS

	<u>Page</u>
STUDY GIST	i
FOREWORD	iii
1. INTRODUCTION	1 - 1
1.1 Background	1 - 1
1.2 Objectives	1 - 1
1.3 Technical Approach	1 - 1
1.4 Organization of this Report	1 - 2
2. FUNCTIONAL AND COMPARATIVE ANALYSES	2 - 1
2.1 Background to the MAMP Database Transition	2 - 1
2.2 Initial Review of the MAMP Database	2 - 1
2.3 Subsequent Modification	2 - 7
2.4 Database Integration Process	2 - 7
3. EXTENSIONS TO THE CSS/EMW/SOF MAMP	3 - 1
3.1 General	3 - 1
3.2 BRDEC MAMP Reports	3 - 1
3.3 MAMP Program Prioritization Technique	3 - 1
3.4 MAMP Relationship to the LRRDAP	3 - 4
4. CONCLUSIONS AND RECOMMENDATIONS	4 - 1
4.1 Conclusions	4 - 1
4.2 Recommendations	4 - 2
APPENDICES	
A. MAMP DATABASE STRUCTURE	
B. MISCELLANEOUS PROCEDURES	
C. SAMPLE OUTPUTS OF CURRENTLY AVAILABLE MAMP REPORTS	
D. SOURCE CODE FOR CURRENTLY AVAILABLE MAMP REPORTS	

Section 1

INTRODUCTION

1.1 Background

The automated Mission Area Materiel Plan (MAMP) has been under development since December 1984. As a joint AMC/TRADOC program, the MAMP is intended to relate the materiel development initiatives of AMC to the TRADOC developed Battlefield Development Plan (BDP) deficiencies. The objective of the MAMP is to more closely align ongoing and planned Army RDT&E activities with stated Army requirements. The automated MAMP system supports this objective with responsive and comprehensive reports of RDT&E program plans and through consistent application of prioritization techniques to relate the RDT&E to the BDP.

In June 1985, AMC adopted the initial work performed at BRDEC on the automated Combat Service Support (CSS) MAMP and began the process of transforming it to be applicable to all mission areas. AMC provided the first release of its software to the MAMP community in January 1986. At the same time, the mission area assignments were realigned to be consistent with the TRADOC mission areas (when earlier they were based on the DA mission areas in use by AMC). BRDEC was assigned to represent the CG, TROSCOM in his role as Mission Area Manager of the Combat Service Support (CSS), Engineering and Mine Warfare (EMW), and Special Operations Forces (SOF) mission areas.

1.2 Objectives

The objectives of this effort were to provide management science support to test, analyze, and apply the revised AMC MAMP software; to identify and implement extensions to the MAMP that address BRDEC unique requirements; and to further develop alternative methods to integrate the program prioritization and funding allocation processes.

1.3 Technical Approach

These objectives were accomplished through the following efforts:

(1) Functional and comparative analyses of the original BRDEC and revised AMC MAMP software were performed to identify all substantive differences in their content and performance. In many areas, the AMC MAMP system was an improvement over the original BRDEC database. The differences were such that it was determined that the BRDEC MAMP would be made to conform structurally with the AMC MAMP.

(2) Extensions to the MAMP software were added so that the MAMP would be applicable to the BRDEC unique requirements that span across several TRADOC Mission Areas. This involved revising all existing BRDEC MAMP reports to conform to the AMC database structure, and particularly the AMC MAMP control files. All new MAMP reports generated in this effort were designed to generate output appropriate to BRDEC as well as to the designated TRADOC Mission Areas.

(3) Other program prioritization techniques, specifically those that integrate the MAMP into the DA LRRDAP process, were reviewed and implemented to assist in the assessment of reprogramming recommendations.

1.4 Organization of this Report

This technical report presents a complete synopsis of the work that was performed under this Task Order.

Section 2 describes the functional and comparative analyses that were performed and documented in an Interim Report. In addition, it includes the functional analysis of the MAMP database integration process and the text of the recommendation made to AMC to correct inherent defects in that process.

Section 3 documents the techniques used to implement the BRDEC unique reports and the advanced program prioritization algorithms. Examples of these techniques can be observed in the Appendices.

Section 4 presents the findings and recommendations of this study effort. These are focused on the additional actions necessary to continue the development of the MAMP process and to improve its utility to BRDEC.

Appendix A documents the BRDEC MAMP database structure. It contains several unique data files not included in the AMC database.

Appendix B documents the miscellaneous procedures (UNIX shells and C programs) needed to correctly manage the BRDEC MAMP.

Appendix C presents samples of each of the BRDEC MAMP reports, along with instructions for running those reports.

Appendix D provides a record copy of the source code for each of the ACE reports contained in the BRDEC MAMP system.

Section 2

FUNCTIONAL AND COMPARATIVE ANALYSES

2.1 Background to the MAMP Database Transition

In early 1986, BRDEC received the first release of the AMC generated MAMP system. This effort was a major restructuring of the MAMP system software from its configuration in 1985. It incorporated many of the features of the RDA database system, and of the earlier "mat_plan" database and Informix ACE reports used at BRDEC. It added a menu structured user interface to enable users with less computer knowledge to operate the system and data entry forms to reduce the likelihood of data entry errors.

The new MAMP database posed a challenge to the existing CSS/EMW/SOF MAMP system. First, our database needed to be revised from the "mat_plan" structure to the "mamp" structure imposed by AMC. This would allow the convenient exchange of data among the various MAMP sites. In addition, our existing ACE reports needed to be revised to accommodate the changes to the database structure and data elements.

2.2 Initial Review of the MAMP Database

An initial review of the revised MAMP database led to the MRC Interim Report of May 1, 1986, which is reproduced at Figure 2-1. It reported that the new "mamp" database was an improvement over the old "mat_plan" database, but that the total system design was still not well thought through. In particular, it pointed out that many of the new system features, like the "perform" data entry screens, were particularly difficult to use and would be unsuitable for our requirements. It was also noted that the new "mamp" database structure would not allow for all the existing and contemplated reports for the CSS/EMW/SOF MAMP.

During the preparation of this report, the CSS/EMW/SOF MAMP database was transitioned in its entirety to be compatible with the AMC MAMP software. This meant that all common data elements between the "mamp" and the "mat_plan" databases were brought into conformance with the "mamp" database. Any additional data elements not included in the AMC standard "mamp" database were added to "mamp" as separate files so as not to corrupt the standard file definitions. Subsequently, all existing CSS/EMW/SOF reports were revised to utilize that new structure.

INTERIM REPORT ON THE MAMP DATABASE

Prepared By: Neil Romstedt
McLean Research Corporation
1 May 1986

A. OBJECTIVE

This interim report is provided in accordance with Subtask 1 of Task Order 0016 of DAAK70-84-D-0052. It documents the preliminary findings of this contractor in the use of and transition to the AMC revised mamp database and software.

B. DATABASE DIFFERENCES

The mamp database is not significantly different from the mat_plan database. Both contain approximately the same data about systems, workpackages, and deficiencies. The major difference between the two is the organization of funding data for workpackages, tasks, and projects. While mat_plan uses a single record to include the funding stream over the eight year horizon of the planning data, the mamp uses individual records for each of the years in which funding data is provided. The mamp technique reflects the database structure used in the RDA budget database, from which the data is derived. This technique has advantages and disadvantages. The primary advantage is that if funding for any particular year is blank, then the record does not have to be entered in the database. In the mat_plan database, the field is reserved and set to 0.0 if no data is entered. The primary disadvantage is that multiple records must be used for any one workpackage, task, or project. Each record has overhead in terms of the repeated information needed to identify and link the record to the other associated data. In general terms, if the overhead can be kept small, then it is usually preferred to have separate records. However, the complexity of the data in the MAMP dictates that there be a large quantity of overhead data.

For example, to correctly link a specific funded, unfunded, and guidance quantity for a workpackage for one year, the mamp database requires the workpackage command, category, subcategory, pe, project, task, workpackage number, and year. These data total 50 bytes per record (out of about 80 total bytes). In addition, the index overhead of about 35 bytes per record must be included. If four years of data were entered, then the four records would consume a total of about 460 bytes. If the data were organized as a single record of eight years, then the overhead would be 44 data bytes per record (out of about 290 bytes total) and the index overhead would be about 20 bytes, for a total of 310 bytes. Some of this disparity could be eliminated through better database design practices to eliminate unnecessary overhead data, and this action should certainly be taken. The potential future expansion to an Extended Planning Annex (EPA) forecast of an additional 8 years (16 years total) of RDT&E funding could clearly alter the balance in the favor of individual records, so this issue should be carefully considered.

In addition, the organization of the data into individual years presents some intriguing and challenging problems to the casual ACE report writer. The level of complexity of the report writing problem multiplies by the number

Figure 2-1. Interim Report on the MAMP Database

of individual data records which must be handled. Without clever programmers and thorough training, the reports quickly become so complex and the files produced so large that the user is deterred by energy and time from using the database to solve his unique problems.

Both the mamp and mat_plan databases suffer from being not well thought out and designed. The mamp database deserves our attention since it is the database that AMC has designated and will use. There are several design improvements that could be suggested at this point, but since a thorough redesign is appropriate no recommendation is made at this point. Some issues to consider in the redesign are:

(1) What is sufficient and unique to identify a workpackage? Does the identity chain include command, category, subcategory, pe, project, task, and workpackage number? Or is the command, category, and workpackage number sufficient? I believe that the command, pe, project, task, and workpackage number are enough, and that the category and subcategory of any particular project should be maintained in the project file.

(2) What does the acquisition code have to do with the linking of systems to deficiencies and why is it in the ssndef file?

(3) Why is the priority of a particular pdip contained in the lrrdp file? This allows careless data entry clerks to put different priorities in for the same pdip/increment. The pdip priority, along with any other vital data about the pdip that may be required should be in a pdip file.

(4) Workpackage contribution values to systems that they are linked to need to be included in the wkpsys file. This change is critical, and has already been done (unilaterally) for our mamp database.

(5) The project index field in the project control file needs to be switched to allow duplicate entries for any one project. This will allow one project to be used against different mission areas, commands, or flags. Currently the schema will allow only one entry for any one project. This same problem is present in the workpackage control file. Both have been changed (unilaterally) for our mamp database.

(6) The system title should be in the ssn file and not the lrrdp file since it is the same for each pdip/increment that a system may be linked to. This change will make the ssn file, not the lrrdp file the key file entry point for systems.

(7) Why can't the workpackage funding data and the workpackage statement data be combined into a single file? Both use exactly the same structure for identifying the relevant workpackage and year concerned. Combining them will eliminate half the overhead of the two files. This might conflict with a decision to merge the funding data into a single record.

(8) What purpose is the workpackage control file that is not served by the project control file?

(9) The workpackage description file could be merged into the wkpkg file. This would create a long workpackage record but would cut the overhead by half.

Figure 2-1. Interim Report on the MAMP Database (Continued)

(10) The classification identification procedures that we use in the mamp are not very thorough. It is difficult to tell, except for the text descriptions, what of the data presented in any of our reports is SECRET or CONFIDENTIAL. The data fields used in the data base to tell us the classification of the data are ambiguous as to which data is to be marked. We need to rethink this process and set some specific standards that we will follow.

C. PERFORM SCREENS

The concept of using perform screens to simplify the operator interface for data entry and application is well understood. In general, the average computer user cannot be trusted to correctly type in data and is usually not well enough versed in a computer's operating system to execute programs without simple and straightforward instructions. The perform screen is intended to provide a common user interface, which with some brief training will allow the data entry clerk and the database administrator alike to access and use the database. This cannot completely correct for erroneous data but using a systematic system like the perform screens can catch a significant fraction of the mistakes that might and probably would occur if the data were entered directly into the database (using enter2).

The perform screens exact a performance degradation on data entry. In well designed screens this penalty will be negligible. However, in the screens provided by AMC, the penalty is terrible. It takes about four times as long to enter data using the perform screens as compared to enter2. Then as an extra little kicker, to exit the perform screens and correctly update the data base, the AMC software forces the user to execute some hidden processes that check to see that all the software defined codes are correct. This process takes about 30 minutes and completely ties up the user's terminal. Furthermore, it also completely ties up the database so that other users cannot gain access to the database to enter their own data. From our experience, the use of the existing perform screens is a huge waste of time. Although it may be appropriate for an occasional user, it is not appropriate for the massive data entry and short-fuzed reporting requirements that the MAM faces.

The perform screens should be used with caution until they can be fixed. This fix should be prepared along with the database redesign discussed earlier.

D. AMC PROVIDED REPORTS

The AMC provided reports are a reasonable, but incomplete, clones of the reports that were being used at Belvoir in the summer of 1985. Since that time the Belvoir reports have changed significantly in content and quantity. The AMC System Summary, Deficiency Summary, and Commodity Summary are adequate but slightly confusing. The use of the control codes and control names to access the desired systems, deficiencies, and commands is not well documented and not intuitive. We were initially unable to reliably get reports out of the AMC system - probably because we did not understand it. The AMC software appeared to produce unpredictable results, like not producing a System Summary for all systems owned by Belvoir when the report is asked for using the menu structure. Obviously, either the wrong questions were asked or the wrong answers provided. This will become very frustrating for other users who have no recourse to run their own reports.

The AMC reports are a little faster than our own. I would guess about

Figure 2-1. Interim Report on the MAMP Database (Continued)

20 percent faster. But these reports also tie up a user's terminal so that no other work can be done while the reports are being run. Typically the user will select output to the printer, since most people need printed reports, but if the printer breaks down during the print job (a frequent occurrence here) then the user is forced to rerun the whole report. The AMC reports appear to have no capability to set the starting page for any one report. Thus, we could not get the reports automatically page sequenced for our MAMP volumes.

Fortunately, we do not have to rely on the performance of those reports. Our own ACE reports are completely within our control to modify and use. We should continue to use and develop them as our needs change. We should offer those reports to ALMSA and other MAMs for their consideration.

E. CHANGEOVER TO THE MAMP DATABASE

The changeover in data from the mat_plan to the mamp database has been completed. The process of moving the data took approximately 16 hours, but much work in scrubbing the data that is there remains. In moving the mat_plan data, some of it was corrupted by the AMC provided integrated database. This problem is discussed in detail in paragraph F below.

The changeover in reports to the mamp database is proceeding more slowly. I have moved 8 old reports and created 5 new ones in the last week. There is still lots more to do in this regard. The changeover requires preparing some temporary database files, going through the programs and changing the variable names, adding the control code as well as the control name, and rechecking all the linking between files to ensure that it is being done properly. I estimate that the reports will all be transferred within the next three weeks. The priority will be given to the reports that constitute the MAMP document.

F. DATA FLOW AND TRANSFER

The data flow process for the integrated MAMP is not well thought through. There is the significant likelihood that there will be circular data flow, so that bad data propagates throughout the system and that errors cannot be reliably corrected. This will occur if the AMC data integration process can not distinguish between scrubbed data from a MAMP or RDA site and data that was not checked by a MAMP or RDA site because it was not relevant to that data manager. Thus, our data errors will be picked up by all other MAMs, and returned to us at some later date in the circular flow, after we have hopefully found and corrected those errors in our own database.

One solution to this problem that we can implement at Belvoir is to maintain a physically separate "integrated" database on our Plexus. There will be room for this when the mat_plan is retired to tape. Then, we will read the AMC tapes into this other database, and carefully control the transfer of data from the integrated database into our mamp database. All of our corrections and data submissions will come from our mamp database to ensure that we are not contributing to the circular data flow.

This approach will also allow us to remove from our mamp database any data that is obviously not appropriate to our mission. If we can get rid of the data that we do not use, then our reports will run faster and our submissions to AMC will not include some other MAMs bad data.

We should also carefully consider the impact of unilateral database changes

Figure 2-1. Interim Report on the MAMP Database (Continued)

on our capability to feed the AMC data submissions. At this point, the two changes we have made will require some special attention in preparing the tape. These changes were critical so they were made, but the decision to make additional changes will magnify the compatibility problems we will likely face.

G. NEW INFORMIX RELEASES

Relational Database Systems has released several new products with the Informix system. The basic Informix package now includes SQL, their version of the IBM structured query language. They have an advanced report writer called 4GL for 4th generation language that they sell as an option.

Another intriguing option is the optional C programming tools package that they sell (Informix-ESQL). These tools consist of a subroutine library that any C program can call to access the database and index files. It appears that any competent programmer can directly access the Informix database from a C program to produce reports (or whatever) without having to write special drivers or use the proprietary ones developed by IITRI.

G. CONCLUSIONS

We are now using the AMC mamp database as best we can. The process of moving our established reports is underway. Uninterrupted attention is required to ensure the success of this venture between now and June.

2.3 Subsequent Modifications

Since its initial release, the AMC MAMP system has undergone four revisions. Each revision has added new database fields and/or files, new or modified reports, and appropriate menu entries and data entry forms to support its enhanced capabilities. To its great credit, the AMC installation instructions have been very thorough and the system has worked exactly as advertised with each revision.

The installation of each revision takes approximately four to eight hours of computer time. The installation process consists of reading in the new release tape, positioning the files in the appropriate directories, and running the installation shell programs. These shell programs automatically update the old database files to their new structure, create new files, add any default data values, revise the menus and perform screens, and recompile and relink the C programs that support the system. During this period the computer is unavailable for other MAMP work. In addition, the changes to the database structure must be carefully examined to determine their impact on the CSS/EMW/SOF MAMP reports. Whatever changes are present must be completely integrated into our database to keep it in conformance with the AMC structure and yet still allow the running of our own reports. This transition imposes an additional workload of approximately four to ten days, depending upon the extent of the required changes to the ACE reports.

2.4 Database Integration Process

One of the goals of the MAMP system is the convenient and rapid exchange of data among the MAMP sites and AMC. An "integrated" database is envisioned where data entered in at one site is available to all users. Because there are many different sites, and they use physically separate computers which are not electrically interconnected, the data used at each site is actually quite distinct. Periodically, the data must be collected and joined together into a single, larger database. This process is initiated with a data tape submission from each MAMP and RDA site to AMC. AMC then consolidates the data and returns the integrated database to the MAMP sites to replace the earlier data. Then, at least until the data is changed by the Mission Area Managers, the MAMP sites will all be using the same data.

The problem of circular data flow that is inherent with this technique was described in the Interim Report. In fact, for the integrated database that was distributed in June 1986, approximately one month of productive effort was lost to the CSS/EMW/SOF system because the integrated database required extensive review and corrective action to restore it to its original status. Bad data that was introduced into the MAMP database during the integration process had corrupted the scrubbed data in the CSS/EMW/SOF system. This bad data could only be identified through close examination and comparison with reports produced under the pre-integration database.

This bad experience led to the BRDEC proposal in July that AMC adopt a more controlled procedure for the database integration process. The text of the proposal, with a detailed procedure that might be followed, is presented at Figure 2-2. This proposal was immediately adopted by AMC, and implemented in August throughout the MAMP system. Its actual effectiveness at correcting the circular data flow problem will be determined at the next data submission.

PROPOSAL FOR MAMP DATABASE INTEGRATION

A. OBJECTIVE

The objective of this paper is to propose a procedure which will greatly speed the process of MAMP database integration and provide the greatest assurance possible that the integrated database will contain only quality data.

B. BACKGROUND

One of the problems yet to be faced by the MAMP process is establishing a structured data flow. Responsibilities for data, its accuracy and its completeness have not been assigned to anyone. At present, we appear to be operating on the principle that more data is better, regardless of where it comes from. On the face of it, this principle is flawed. The database should contain only quality data that has been scrubbed and approved by a responsible authority. Otherwise, we are wasting our time in a "garbage in - garbage out" paper production effort.

Each MAM administrator spends most of his time, and most of his staff's time, paying attention to the details of the data he uses for his MAMP. The database is constantly being updated as new information is developed, systems and programs are built, deficiencies and contribution values are assigned, and data presently in the database are scrubbed.

The MAMP database integration, as it occurred in March-April and May-June, aggravates the problem of data quality. In March, we all sent in data tapes to AMC containing the best data we had at that time. Some of that data was incomplete and inaccurate. In April, we received the consolidated data with everyone's best data as of March. Naturally, we continued to focus on and improve the data pertinent to our own MAMPs and we ignored the data that was not pertinent. In May, we sent in another data tape to AMC for integration. In June, we got back the integrated database but it included all the bad data that we had just spent the last two months cleaning out of our database!

The reason for this is that just as the CSS/EMW/SOF MAM administrator spent no time trying to clean up the FS MAMP data (or anyone else's for that matter), so the FS MAM administrator spent no time on CSS/EMW/SOF data. The result of the integration process is to perpetuate bad data throughout the system. Instead of being identified and eliminated, the bad data looks just like good data and so is redistributed to all MAM sites. This chaotic situation has to be stopped at the time of database integration. The next AMC data call is planned for 21 July and we must be prepared to implement a revised procedure immediately.

Two procedures are proposed to ensure only the best data is integrated into the AMC database. First, only scrubbed data should be sent to AMC by each MAM site. If the MAM administrator is not responsible for that data, then he should not send it in. Second, AMC should not accept data for which the MAM administrator is not responsible and should establish automated means

Figure 2-2. Proposal for MAMP Database Integration

to identify and reject that data.

C. CONTROL FILES

The correct mechanism for establishing what data the MAM administrator is responsible for is through the control files. Control files are maintained in the MAMP database for Deficiencies (defctl), Systems (ssnctl), Projects (prjctl), and Workpackages (wkpctl). The control files allow the MAMP software to automatically select the specific data out of the database that is pertinent to any one MAMP.

The control file structure is a bit difficult to grasp at first, but it makes sense when you understand that it can be used for a wide variety of control functions and that the same structure is consistently applied in all of the control files.

The control files use the field "code" (dcl_code, scl_code, pcl_code, and wcl_code) to denote whether the record refers to a TRADOC Mission Area, a DA Mission Area, an MSC, or a Flag. Thus, in the same file, a single system could be designated as being included in a TRADOC mission area (EMW), a DA mission area (CSS), a command (BELVOIR), and a flag (LOG R&D) MAMP. This gives the database the flexibility to be used for other purposes than just the current TRADOC Mission Areas of interest today. If a MAM site cares to have multiple MAMPs in a single database then the control files provide a way to keep them straight. At the moment, our concern and responsibility is only for the TRADOC Mission Areas that have been assigned. They are all designated by using the code = 1.

The control file field "name" (dcl_name, scl_name, pcl_name, and wcl_name) are used to separate out the Mission Areas, Commands, or Flags, depending upon the value of the "code" field. So at Belvoir, we use "CSS", "EMW", and "SOF" in the "name" field. The field has been set to twelve (12) characters wide to accommodate the maximum size of a command name (should the code = 3), although all the TRADOC Mission Areas are two or three characters wide.

So, if the FS MAM administrator is using data in the database, then his deficiencies, systems, projects, and workpackages are identified by corresponding records in the control files with code = 1 and name = "FS". This is the same criteria that should also be used to automatically select out the data that should be sent to AMC in a data call.

D. SPECIFIC FILES TO SEND AND PROPOSED PROCEDURES

The following paragraphs will discuss each file that should be sent, and the criteria that should be used to select the specific records to send.

The Informix command "Informer" should be used to select these records. The UNIX/Informix commands to do this are shown for each file. To get started type:

```
cd                (to get you to the home directory)
cd data          (or wherever you want the output files)
rm -i *          (to get rid of all the old .uld files)
informer mamp
```

Figure 2-2. Proposal for MAMP Database Integration (Continued)

I have used the CSS mission area to illustrate the process. Just substitute your own mission area name in for CSS wherever it is shown.

1. SSNCTL: The System Control File. Systems are selected in conjunction with the combat and materiel developer, appropriate for the mission area.

unload ascii ssnctl where scl_code = 1 and scl_name = "CSS"
to "ssnctl.uld";
2. DEFCTL: The Deficiency Control File. Deficiencies are assigned to MAMs.

unload ascii defctl where dcl_code = 1 and dcl_name = "CSS"
to "defctl.uld";
3. PRJCTL: The Project Control File. Projects are assigned to MAMs.

unload ascii prjctl where pcl_code = 1 and pcl_name = "CSS"
to "prjctl.uld";
4. WKPCTL: The Workpackage Control File. This file is optional. It is used to identify specific workpackages in a mission area when the entire project cannot be identified using the project control file. This would occur if the project were shared among two mission areas.

unload ascii wkpctl where wcl_code = 1 and wcl_name = "CSS"
to "wkpctl.uld";
5. DEF: The Deficiency Description File. The assigned deficiency manager should be responsible for any updates and corrections to the deficiency data.

unload ascii def where dcl_code = 1 and dcl_name = "CSS"
joining def_def = dcl_def to "def.uld";
6. SSNDEF: The System to Deficiency Linkage File. The assigned deficiency manager should be responsible for all entries against his deficiencies. The system manager can nominate linkages to deficiencies, and enter unrated linkages into the integrated database, but must coordinate with the deficiency manager to have a contribution value assigned and entered into the database.

(find linkage for all mission area deficiencies)
read into a ssndef where dcl_code = 1 and dcl_name = "CSS"
joining ssndef_def = dcl_def;

(find linkage for all mission area systems)
read into b ssndef where scl_code = 1 and scl_name = "CSS"
joining ssndef_ssn = scl_ssn;

(find MA systems linked to non-MA deficiencies)

Figure 2-2. Proposal for MAMP Database Integration (Continued)

```

assign c = b minus a;

      (blank out the contribution values for these linkages)
read into d c.ssndef_ssn c.ssndef_def cv=" " c.ssndef_acq_code;

      (add together the two linkages of interest and unload)
assign e = a union d;
unload ascii e to "ssndef.uld";

```

7. LRRDP: The System to PDIP Linkage File and Procurement Funding Stream. The system manager will enter in new records into the lrrdp file for all systems assigned NSIs and for systems with SSNs that were forgotten in the DA LRRDAP. The funding data for these additional entries will be all zeros. AMC will ensure that the latest LRRDAP funding data and PDIP alignments and priorities are reflected in the integrated data base.

```

unload ascii lrrdp where scl_code = 1 and scl_name = "CSS"
joining lrrdp_ssn = scl_ssn to "lrrdp.uld";

```

8. SSN: The System Description File. The system manager will enter in and correct system description data.

```

unload ascii ssn where scl_code = 1 and scl_name = "CSS"
joining ssn_ssn = scl_ssn to "ssn.uld";

```

9. WKPKG: The Workpackage Definition File. The workpackage manager will enter in or revise the workpackage to PDIP linkage into this file. Workpackages are identified by belonging to controlled projects and/or by being controlled workpackages themselves.

```

      (find all workpackages that are part of controlled projects)
read into a wkpkg where pcl_code = 1 and pcl_name = "CSS"
joining wkpkg_proj_idx = pcl_idx;

      (find all workpackages that are individually controlled)
read into b wkpkg where wkp_code = 1 and wkp_name = "CSS"
joining wkpkg_no_idx = wcl_idx;

      (join the workpackages together and unload)
assign c = a union b;
unload ascii c to "wkpkg.uld";

```

10. WKPSYS: The Workpackage to System Linkage File. The MAM will enter in or revise the linkages and include criticality factors as needed. The system manager will be the sole authority to assign the criticality factors. The project/workpackage manager will nominate linkages for review by the system manager. Note that this file should completely replace the mamp file in the RDA database.

```

      (find linkages for all mission area systems)
read into a wkpsys where scl_code = 1 and scl_name = "CSS"
joining wkpsys_ssn = scl_ssn;

```

Figure 2-2. Proposal for MAMP Database Integration (Continued)

```

      (find linkages for all mission area workpackages)
read into b wkpsys
joining wkpsys_wkpkg_idx = c.wkpkg_wkpsys_idx;      (using c above)

      (find all MA workpackages linked to non-MA systems)
assign d = b minus a;

      (blank out the criticality factors for these linkages)
read into e d.wkpsys_idx cf=0;

      (join together the two files of interest and unload)
assign f = a union e;
unload ascii f to "wkpsys.uld";

```

E. CONTROL REQUIREMENTS FOR INTEGRATION

AMC should take care to ensure that the principles applied here are followed throughout the integration process. Particular emphasis should be given to the responsibilities for data input from each participant, so that a MAM administrator will contribute only the data for which he is responsible. Even so, AMCs automated database integration mechanisms should not accept data that does not conform to these principles:

1. Deficiency related data comes from the deficiency manager only.
2. System related data comes from the system manager only.
3. RDTE related data comes from the MSC through the RDA database only.
4. Other workpackage data comes from the RDTE project manager only.

There are bound to be conflicts in the data records, even after conforming to these principles. Conflicts in linkage files should be resolved in favor of the deficiency manager and then the system manager. Conflicts in system files should be resolved in favor of the designated TRADOC proponent for that system (as indicated in the `ssn_tradoc_pro` field).

F. FUTURE CONSIDERATIONS

AMC should give some consideration to the eventual need for a PDIP control file, since PDIPs are assigned to TRADOC proponents and the direction of our future efforts will be more closely aligned with the PDIP structuring process.

Figure 2-2. Proposal for MAMP Database Integration (Continued)

Section 3

EXTENSIONS TO THE CSS/EMW/SOF MAMP

3.1 General

The work required by this Task Order directed the evolving MAMP toward addressing three new areas. First, the existing MAMP reports were modified and the database structures established to allow their application to BRDEC management tasks. Second, the MAMP program prioritization techniques were incorporated into a series of new reports which show the critical fiscal reprogramming decisions which might be made, and what their likely impact would be on the executability of RDT&E programs. Finally, these efforts culminated in a second new series of reports, which depict the consequence of DA LRRDAP priority decisions and which institute the MAMP prioritization of RDT&E programs into the PDIP submission process.

3.2 BRDEC MAMP Reports

In keeping with the revised MAMP database structure, it is now possible to designate command names as the control file filters. The significance of this is that every MAMP report can be produced for BRDEC with no additional programming required. What is needed is an up-to-date and complete list of all systems, projects, workpackages, and deficiencies of interest to BRDEC. This list must be maintained within the database, just as are similar lists for the CSS, EMW, and SOF mission areas. Preliminary BRDEC control file lists were generated in June and these have sufficed without modification to date. However, this data should be reviewed and updated if necessary to ensure that the correct MAMP data is being represented as pertinent to BRDEC.

3.3 MAMP Program Prioritization Technique

3.3.1 The Existing Prioritization Technique

The methodology employed to rank order systems and workpackages in the MAMP has been directed by TRADOC. It computes a score for a system based on the Battlefield Development Plan (BDP) deficiencies it supports and its assigned contribution value to the solution of each of those deficiencies. While the linkage between systems and deficiencies may be proposed by the AMC materiel developer, the contribution values are assigned by the TRADOC mission area representatives. Values from A to E (and X) are assigned. The TRADOC representative is instructed to use no more than one A rating for each deficiency. A geometric progression is used to translate the letter value into a numerical score, as follows:

<u>Letter</u>	<u>Weight</u>
A	16 points
B	8
C	4
D	2
E	1
X	0

The deficiencies themselves are also weighted according to their numerical sequence. This is in keeping with the TRADOC rationale for placing the BDP deficiencies in numerical order. BDP deficiency number 1 is more important than deficiency number 2, and so on. The mathematical representation which describes the relative value of the deficiency is:

$$\text{Deficiency Value} = (500 - \text{Deficiency Number}) / 500$$

This places the deficiency value in the range between 1.0 and 0.0 (so long as the deficiency numbers do not exceed 500). Other formulations have been proposed which have provided a greater relative value to lower deficiency numbers, but these have been shown to have only a marginal impact on the system rank ordering results obtained in actual test samples.

A system score is computed as the product of the deficiency value times the contribution weighting, summed over all deficiencies it supports:

$$\text{System Score} = \sum \text{E Contribution Weight} \times \text{Deficiency Value}$$

The higher the system score, the more important it is to the solution of the Army's BDP deficiencies.

Workpackage scores are computed in a similar fashion, building on the computed system scores. Each workpackage is rated as to its significance to the fielding of a system. The AMC materiel developer assigns this criticality factor, a number 1 through 5, with the limitation that a workpackage can be rated a 1 (most critical) against only one system. The weightings associated with each factor are as follows:

<u>Factor</u>	<u>Weight</u>
1	10 points
2	5
3	2.5
4	1.25
5	0

The workpackage score is computed as the product of the system score times the criticality weighting, summed over all systems it supports:

$$\text{Workpackage Score} = \sum \text{E Criticality Weight} \times \text{System Score}$$

The higher the workpackage score, the more important it is to the development of systems which solve BDP deficiencies.

3.3.2 Criticism of the Existing Prioritization System

The above technique for rank ordering systems and workpackages has received a lot of valid criticism. In general terms, this criticism focuses on the following aspects:

(1) The BDP itself is insufficient. It has an inherently short range of view of Army needs and is thus inappropriate for application to all tech base activities. It focuses entirely on the tactical battlefield at Corps level and below, and thus entirely ignores the Army's significant activities in the theater communications zone, in the Continental United States, and in strategic defense. The deficiencies themselves are rank ordered in a quasi-political process which raises doubts as to the objective value of the ranking or its suitability for long term RDT&E planning.

(2) The computation technique is flawed. Some other scheme should be used to translate the deficiency numbers, contribution values, and criticality factors into a quantitative measure of the virtue of a system or workpackage. In particular, the existing technique gives too much value to systems which support multiple deficiencies and to workpackages which support multiple systems, with too small regard for the importance of that support.

(3) The assignment of contribution values and criticality factors is flawed. The control process for assigning these values and factors is too easily circumvented, and thus the system and workpackage scores can be easily manipulated to the detriment of competing programs.

(4) The resolution of this technique is inadequate. A prioritization scheme based on the BDP, because of its limited scope, is inherently unable to make fine distinctions between competing programs. It is only suitable for coarse comparisons of program value. A knowledgeable manager must review the results and apply his expert judgement to adjust the computed rank ordering. It is difficult to imagine any scheme based on reasonable and economically obtained data that will supplant this managerial function.

3.3.3 Potential for Enhancement

There are numerous improvement efforts which might be undertaken to address these criticisms. In particular, a revised computational technique might be proposed which uses the same input data, yet eliminates some of the bias of the method toward systems which support multiple deficiencies and workpackages which support multiple systems. Any revised technique would have to be demonstrated to TRADOC and be approved by them for use in the MAMP system.

Attempting to make the methodology more comprehensive is probably unworkable. The BRDEC experience with the "goal programming" method illustrates the actual difficulty in obtaining objective data for use in the method, and the (un)satisfying realization that after all the systematic analysis is performed, human managerial judgement cannot be dispensed with.

Instead, the role of the knowledgeable manager could be reinforced by having the MAMP system provide a ready source of data in a consistent format. This will not be an easy task, because no small amount of data can adequately represent the virtue or complexity of a program effort. The current BRDEC briefing presentation format for programs (FOE matrix, executability charts, milestone charts) is an appropriate method for illustrating and initiating the discussion of substantive matters. Specialized MAMP reports should be tailored to match these standard formats and to provide backup materials should more detailed discussions be required. The existing reports which illustrate the impact of key fiscal reprogramming actions can be enhanced and employed as a decision aid.

3.4 MAMP Relationship to the LRRDAP

The relationship of the MAMP to the DA Long Range Research Development and Acquisition Plan (LRRDAP) has not yet been formalized. But, the MAMP has been directed to proceed in the direction of supporting the LRRDAP process. Much of the recent innovative work in the MAMP has focused on developing and evaluating usable tools for this effort. This effort is not yet complete. The reports which have been developed need additional definition and refinement to ensure that they are responsive and accurate.

The responsibility for submitting materiel development input to the LRRDAP rest with TRADOC. In preparation of Program Development Increment Packages (PDIPs), TRADOC consults with AMC to establish the RDT&E schedule and resource requirements for the materiel development initiative. The process of preparing requirements documents independently and continuously parallels this effort to ensure that the materiel development programs are responsive to Army needs. In what has been an annual submission to DA, TRADOC develops new PDIPs, revises existing PDIPs if necessary, and establishes its recommendation for the relative priorities for these PDIPs.

DA considers all PDIPs submitted, along with its funding guidance from DoD, and applies its own judgement in the rank ordering of the PDIPs. These are published as the DA LRRDAP. Based on the rank order, a particular PDIP is placed in a funding category (e.g., 'at risk', 'unfunded', 'high priority') to describe its likelihood of actually being funded. The whole of this process is well beyond the limited scope of this report. It is extremely complex, with many external factors that can significantly alter the PDIP structures and priority recommendations submitted by TRADOC.

The key fact is that the MAMP can contribute substantially to the development of PDIPs by providing the detailed information concerning RDT&E programs. As an AMC planning document, the MAMP shows the program schedules and resource requirements as AMC sees them. It forces close AMC coordination with the TRADOC representatives who are charged with preparing the PDIPs. Since it uses the TRADOC prioritization algorithms to rank order the RDT&E efforts, it can be used to predict in advance the relative ranking that TRADOC will assign. In a series of new specialized reports, the MAMP data can be presented in PDIP format, thus relieving the TRADOC action officers of much of the collation and assembly work entailed in preparing the PDIPs.

Other MAMP current reports also depict the consequences of DA PDIP priority decisions. By comparing the DA priorities with the results obtained using the TRADOC algorithm, inconsistencies in the DA funded RDT&E program can be easily spotted. These inconsistencies could be visible in non-executable development programs, or in insufficient funding for programs that address high priority Army needs. The appropriate Mission Area Manager or Commander can use this knowledge to assist him in making reprogramming recommendations, and to visualize the consequences of reprogramming actions.

Section 4

CONCLUSIONS AND RECOMMENDATIONS

4.1 Conclusions

(1) The usefulness of the current MAMP database to the needs of BRDEC is somewhat limited. The database does not currently contain a complete and accurate list of the systems, projects, workpackages, and deficiencies of interest to the command. A preliminary list has been compiled but not validated. "Belvoir control files" need to be maintained, just as are the three mission area control files. This maintenance is not automatic, but requires careful attention and managerial judgement. Without these control files, BRDEC will get relatively little out of the MAMP system in the way of validated and comprehensive reports.

(2) Rigorous procedures need to be instituted across all MAMP sites to ensure that only good data is retained within the system. The actions of AMC in adopting BRDEC's recommendation concerning this issue is laudable, and initial procedures were established promptly. This is indicative of the rapid advance in the level of sophistication and detail of the AMC MAMP software.

(3) Better hands-on operator training in the use of the AMC MAMP software is required before that software will be widely accepted. The existing documentation is inadequate, and since the system software is changing so fast the documentation is unlikely to catch up any time soon. The training should also stress the ability of the MAMP sites to develop their own unique reports to support their unique requirements.

(4) The data contained in the MAMP database requires a lot of work to assemble and maintain. The quality control functions require a significant degree of knowledgeable high level attention. The MAMP system does not invent the data. Instead, human beings type the data into one of the many computers in the system. As a result, the data in the database may not always be correct but it does accurately reflect what was entered into it.

(5) The entire MAMP system (database and reports) structure needs to be rethought from the perspective of the final objectives. The MAMP objectives are to support the development of a LRRDAP responsive to the stated needs of the Army and to rationally allocate available resources to high priority programs. To this end, the PDIP related database structure needs to be stronger, to include PDIP control files, so that tighter fiscal accountability within mission areas can be maintained.

(6) The MAMP prioritization methodology might require additional attention. It is uncertain whether a more detailed and comprehensive methodology would yield any better results. The goal programming model provides a nice contrast to the MAMP technique, and proves that intricate approaches to prioritization have many drawbacks. No matter what technique is employed, AMC decision makers will retain their prerogative to override any computer algorithm to ensure that critical Army programs are continued. Although there are some fundamentally different prioritization schemes that

might be considered more "fair", it is TRADOC's decision as to which methodology to use in the MAMP. BRDEC can explore and recommend alternatives as it sees fit.

(7) The greatest potential for the MAMP is the close cooperation it engenders between the AMC materiel developer and the TRADOC combat developer in establishing goals and recommending programs to DA. In many cases, there has been insufficient communication between these parties. The MAMP provides a formal mechanism and a visible product of that communication which is less demanding than the preparation of requirements documents. As a planning tool, the MAMP excels in visualizing program structure and resource requirements for the future.

(8) The MAMP can also be used to augment the decision making process for allocating and reprogramming committed RDT&E resources. It can recommend actions that are responsive to the BDP, and it can illustrate the consequences of decisions in terms of program executability and fiscal balance.

4.2 Recommendations

(1) BRDEC should establish a specific responsibility for ensuring that the BRDEC interests in the MAMP database are appropriately represented. This should include the specification and maintenance of "Belvoir Control Files" and supervisory quality control functions for BRDEC data.

(2) BRDEC should continue to present recommendations for new reports and procedures to be instituted into the AMC MAMP. AMC has shown a willingness to receive such recommendations and to act upon them. In every case they have extended and improved upon the original recommendation. This should include structural changes to the database to make the system more responsive to the PDIP submission process.

(3) Quality control functions for the MAMP database should be dispersed as much as possible. The directorate/program manager providing the data should be reviewing its own submissions. This may be accomplished by using data entry forms and screens, published examples and standards for quality data, and automated quality control audits. The distribution of MAMP data to the originators for use on personal computers should also be explored as a means to have the data scrubbed.

(4) The specification of a 'fair' prioritization algorithm should be left to TRADOC. The MAMP should be able to employ TRADOC internal procedures for establishing program priority. Extensions or deviations from this algorithm which are responsive to the resource allocation decision within AMC should be pursued and used independently.

APPENDIX A

MAMP DATABASE STRUCTURE

This Appendix describes the file structure of the MAMP database. Files are categorized as either "temporary" or "permanent". Permanent files contain original data, while temporary files are periodically rebuilt from the permanent data to assist in the running of the ACE reports. The files have been further classified according to their function:

<u>Permanent:</u>	Data Type Files
	Control Files
	Primary Data Files
	RDA Database Derived Files
	CSS/EMW/SOF MAMP Specific Files
<u>Temporary:</u>	RDA Funding Rollup Files
	Periodically Updated Files
	ACE Report Preprocessing Files

A.1 Permanent Data Files

A.1.1 Data Type Files

These files contain coded information regarding major data categories of interest to the MAMP system. They should not be modified by an individual MAMP site, since it is critical that the codes all represent the same data meanings throughout the MAMP system. The "acq" file defines what the acquisition code numbers mean (e.g., base case, type classified, development, etc.). The "ctl" file defines what the control code numbers mean (e.g., TRADOC Mission Area, DA Mission Area, MSC, or Flag).

```
file acq                                     permission all

field acq_code                               type integer
field acq_type                               type char 30
field acq_stt                                type char 4

data record length      36
number of records      13
number of indexes      1
acq_code                duplicates not allowed
```

```
file ctl                                     permission all

field ctl_code                               type integer
field ctl_type                               type char 20
field ctl_idx                                type composite
      ctl_code, ctl_type

data record length      22
number of records      5
number of indexes      3
ctl_code                duplicates not allowed
ctl_type                duplicates not allowed
ctl_idx                 duplicates not allowed
```

A.1.2 Control Files

The control files indicate which data is pertinent to each mission area or command. The control files are used to sift the large MAMP database and extract only that data desired for any particular report. Control files currently exist for deficiencies, projects, systems, and workpackages. Each control file has a similar structure, with a code and name field, to allow a specific item to be controlled in more than one mission area or by more than one command.

```
file defctl                                     permission all

field dcl_code                                 type integer
field dcl_name                                 type char 12
field dcl_def                                  type integer
field dcl_idx                                  type composite
      dcl_code, dcl_name, dcl_def

data record length                             16
number of records                             189
number of indexes                              3
dcl_code                                       duplicates allowed
dcl_def                                       duplicates allowed
dcl_idx                                       duplicates not allowed
```

```
file prjctl                                     permission all

field pcl_code                                 type integer
field pcl_name                                 type char 12
field pcl_cmd                                  type char 12
field pcl_cat                                  type char 3
field pcl_subcat                               type char 4
field pcl_pe                                   type char 5
field pcl_proj                                 type char 4
field pcl_idx                                  type composite
      pcl_cmd, pcl_cat, pcl_subcat, pcl_pe, pcl_proj

data record length                             46
number of records                             209
number of indexes                              1
pcl_idx                                       duplicates allowed
```

file ssnctl permission all

field scl_code type integer
field scl_name type char 12
field scl_ssn type char 6
field scl_idx type composite
scl_code, scl_name, scl_ssn

data record length 20
number of records 990
number of indexes 3
scl_code duplicates allowed
scl_ssn duplicates allowed
scl_idx duplicates not allowed

file wkptl permission all

field wcl_code type integer
field wcl_name type char 12
field wcl_cmd type char 12
field wcl_cat type char 3
field wcl_subcat type char 4
field wcl_pe type char 5
field wcl_proj type char 4
field wcl_task type char 9
field wcl_wkp type char 7
field wcl_idx type composite
wcl_cmd, wcl_cat, wcl_subcat, wcl_pe, wcl_proj, wcl_task, wcl_wkp
field wcl_sys_idx type composite
wcl_code, wcl_name, wcl_cmd, wcl_cat, wcl_pe, wcl_proj, wcl_task, wcl_wkp

data record length 62
number of records 1840
number of indexes 2
wcl_idx duplicates allowed
wcl_sys_idx duplicates not allowed

A.1.3 Primary Data Files

These files contain the major MAMP user entered data. They generally define the deficiencies and systems in the database, while workpackage related data is generally derived automatically from the RDA database. The MAMP user should confine his data entry to these files (as well as the control files) as much as possible.

```
file assn                                     permission all

field assn_ssn                               type char 6
field assn_assoc_ssn                         type char 6
field assn_idx                               type composite
      assn_ssn, assn_assoc_ssn

data record length      12
number of records      470
number of indexes       2
assn_ssn                duplicates allowed
assn_idx                duplicates not allowed
```

```
file def                                     permission all

field def_def                               type integer
field def_da_ma                             type char 3
field def_tradoc_ma                         type char 3
field def_type                              type char 16
field def_desc                              type char 600
field def_class                             type char 1

data record length      625
number of records       0
number of indexes       1
def_def                 duplicates not allowed
```


file lrrdp permission all

field lrrdp_ssn	type char 6
field lrrdp_title	type char 60
field lrrdp_pdip	type char 6
field lrrdp_incrpri	type float
field lrrdp_procf1	type long
field lrrdp_procf2	type long
field lrrdp_procf3	type long
field lrrdp_procf4	type long
field lrrdp_procf5	type long
field lrrdp_procf6	type long
field lrrdp_procf7	type long
field lrrdp_procf8	type long
field lrrdp_procf9	type long
field lrrdp_procf10	type long
field lrrdp_procf11	type long
field lrrdp_procf12	type long
field lrrdp_procf13	type long
field lrrdp_procf14	type long
field lrrdp_procf15	type long
field lrrdp_procf16	type long
field lrrdp_sys_idx	type composite
lrrdp_ssn, lrrdp_pdip	

data record length	140
number of records	5169
number of indexes	2
lrrdp_ssn	duplicates allowed
lrrdp_sys_idx	duplicates not allowed

file ssn permission all

field ssn_ssn	type char 6
field ssn_acq_code	type integer
field ssn_class	type char 1
field ssn_desc	type char 600
field ssn_req_doc1	type char 6
field ssn_req_doc2	type char 6
field ssn_req_doc3	type char 6
field ssn_amc_msc	type char 12
field ssn_com_line	type char 10
field ssn_family	type char 10
field ssn_xfuncarea	type char 10
field ssn_amc_mgr	type char 12
field ssn_tradoc_pro	type char 3
field ssn_tradoc_ma	type char 3
field ssn_da_ma	type char 3
field ssn_bdp	type integer
field ssn_real	type integer
field ssn_no_def	type integer
field ssn_sys_sum_idx	type composite
ssn_amc_mgr, ssn_ssn	
field ssn_stream_sum_idx	type composite
ssn_amc_mgr, ssn_acq_code, ssn_ssn	
field ssn_com_sum_idx	type composite
ssn_amc_mgr, ssn_com_line, ssn_ssn	

data record length	696
number of records	2186
number of indexes	5
ssn_ssn	duplicates not allowed
ssn_amc_mgr	duplicates allowed
ssn_sys_sum_idx	duplicates not allowed
ssn_stream_sum_idx	duplicates not allowed
ssn_com_sum_idx	duplicates not allowed

```

file ssndef                                permission all

field ssndef_ssn                           type char 6
field ssndef_def                           type integer
field ssndef_cont_value                    type char 1
field ssndef_acq_code                      type integer
field ssndef_sys_sum_idx                   type composite
      ssndef_ssn, ssndef_def
field ssndef_def_sum_idx                   type composite
      ssndef_def, ssndef_acq_code, ssndef_cont_value, ssndef_ssn
field ssndef_def_cont_idx                  type composite
      ssndef_ssn, ssndef_def, ssndef_cont_value

data record length                          11
number of records                          2558
number of indexes                           5
ssndef_ssn                                 duplicates allowed
ssndef_def                                 duplicates allowed
ssndef_sys_sum_idx                        duplicates not allowed
ssndef_def_sum_idx                        duplicates not allowed
ssndef_def_cont_idx                       duplicates not allowed

```

```

file wkpsys                                permission all

field wkpsys_cmd                           type char 12
field wkpsys_cat                           type char 3
field wkpsys_subcat                        type char 4
field wkpsys_pe                            type char 5
field wkpsys_proj                          type char 4
field wkpsys_task                          type char 9
field wkpsys_wkpkg                         type char 7
field wkpsys_ssn                           type char 6
field wkpsys_srf                           type integer
field wkpsys_idx                           type composite
      wkpsys_cmd, wkpsys_cat, wkpsys_subcat, wkpsys_pe, wkpsys_proj,
      wkpsys_task, wkpsys_wkpkg, wkpsys_ssn
field wkpsys_wkpkg_idx                     type composite
      wkpsys_cmd, wkpsys_cat, wkpsys_subcat, wkpsys_pe, wkpsys_proj,
      wkpsys_task, wkpsys_wkpkg

data record length                          52
number of records                          10398
number of indexes                           4
wkpsys_wkpkg                               duplicates allowed
wkpsys_ssn                                 duplicates allowed
wkpsys_idx                                 duplicates not allowed
wkpsys_wkpkg_idx                           duplicates allowed

```

A.1.4 RDA Database Derived Files

These files are loaded into the MAMP database from the RDA database. This transfer is accomplished periodically by AMC. This implies that the workpackage funding data in the MAMP will always be less than fully current. These files constitute a majority of the bulk of the MAMP database.

```
file actual                                permission all

field act_cmd                             type char 12
field act_cat                             type char 3
field act_subcat                          type char 4
field act_pe                              type char 5
field act_proj                             type char 4
field act_task                             type char 9
field act_yr                              type integer
field act_text                             type char 600
field act_task_idx                         type composite
  act_cmd, act_cat, act_subcat, act_pe, act_proj, act_task
field act_idx                             type composite
  act_cmd, act_cat, act_subcat, act_pe, act_proj, act_task, act_yr

data record length      639
number of records      1699
number of indexes       2
act_task_idx           duplicates allowed
act_idx                duplicates not allowed
```

```
file flag                                permission all

field flag_cmd                             type char 12
field flag_cat                             type char 3
field flag_subcat                          type char 4
field flag_pe                              type char 5
field flag_proj                             type char 4
field flag_task                             type char 9
field flag_wkpkg                            type char 7
field flag_name                             type char 5
field flag_yr                              type integer
field flag_fund                             type long
field flag_wkpkg_idx                        type composite
  flag_cmd, flag_cat, flag_wkpkg
field flag_idx                              type composite
  flag_cmd, flag_cat, flag_wkpkg, flag_name, flag_yr

data record length      55
number of records      11541
number of indexes       2
flag_wkpkg_idx         duplicates allowed
flag_idx               duplicates not allowed
```

```

File plan                                     permission all

field plan_cmd                               type char 12
field plan_cat                               type char 3
field plan_subcat                            type char 4
field plan_pe                                type char 5
field plan_proj                              type char 4
field plan_task                              type char 9
field plan_yr                                type integer
field plan_text                              type char 600
field plan_task_idx                          type composite
plan_cmd, plan_cat, plan_subcat, plan_pe, plan_proj, plan_task
field plan_idx                               type composite
plan_cmd, plan_cat, plan_subcat, plan_pe, plan_proj, plan_task, plan_yr

data record length      639
number of records      6803
number of indexes      2
plan_task_idx          duplicates allowed
plan_idx               duplicates not allowed

```

```

file proj                                     permission all

field proj_cmd                               type char 12
field proj_cat                               type char 3
field proj_subcat                            type char 4
field proj_pe                                type char 5
field proj_no                                type char 4
field proj_title                             type char 60
field proj_class                             type char 4
field proj_declass                           type char 12
field proj_damgr                              type char 4
field proj_amcmgr                             type char 1
field proj_pri                               type integer
field proj_bgt_act                           type char 1
field proj_tradoc_ma                          type char 3
field proj_da_ma                              type char 3
field proj_idx                               type composite
proj_cmd, proj_cat, proj_subcat, proj_pe, proj_no

data record length      118
number of records      716
number of indexes      4
proj_cmd               duplicates allowed
proj_pe                duplicates allowed
proj_no                duplicates allowed
proj_idx               duplicates not allowed

```

```

file task                                     permission all

field task_cmd                               type char 12
field task_cat                               type char 3
field task_subcat                            type char 4
field task_pe                                type char 5
field task_proj                              type char 4
field task_no                                type char 9
field task_corr_proj                         type char 4
field task_corr_task                        type char 9
field task_title                             type char 60
field task_shortt                            type char 16
field task_req_doc                           type char 5
field task_req_date                          type char 4
field task_lab                               type char 12
field task_tradoc_ma                         type char 3
field task_da_ma                             type char 3
field task_idx                               type composite
      task_cmd, task_cat, task_subcat, task_pe, task_proj, task_no

```

```

data record length      153
number of records      2566
number of indexes       3
task_cmd                duplicates allowed
task_no                 duplicates allowed
task_idx                duplicates not allowed

```

```

file wkdesc                                     permission all

field wkd_cmd                               type char 12
field wkd_cat                               type char 3
field wkd_subcat                            type char 4
field wkd_pe                                type char 5
field wkd_proj                              type char 4
field wkd_task                              type char 9
field wkd_wkpkg                             type char 7
field wkd_text                              type char 600
field wkd_idx                               type composite
      wkd_cmd, wkd_cat, wkd_wkpkg

```

```

data record length      644
number of records      3723
number of indexes       1
wkd_idx                duplicates not allowed

```

```

file wkpkg                                     permission all

field wkpkg_cmd                               type char 12
field wkpkg_cat                               type char 3
field wkpkg_subcat                            type char 4
field wkpkg_pe                                type char 5
field wkpkg_proj                              type char 4
field wkpkg_task                              type char 9
field wkpkg_no                                type char 7
field wkpkg_pri                               type integer
field wkpkg_amcpri                            type integer
field wkpkg_lab                               type char 12
field wkpkg_title                             type char 60
field wkpkg_shortt                            type char 16
field wkpkg_trans_date                        type char 4
field wkpkg_da_ma                             type char 3
field wkpkg_tradoc_ma                         type char 3
field wkpkg_systems                           type integer
field wkpkg_pdip                              type char 6
field wkpkg_class                             type char 1
field wkpkg_amc_adj                            type long
field wkpkg_mam_adj                           type long
field wkpkg_score                             type double
field wkpkg_ranking                           type long
field wkpkg_proj_idx                          type composite
wkpkg_cmd, wkpkg_cat, wkpkg_subcat, wkpkg_pe, wkpkg_proj
field wkpkg_wkpsys_idx                        type composite
wkpkg_cmd, wkpkg_cat, wkpkg_subcat, wkpkg_pe, wkpkg_proj, wkpkg_task, wkpkg_no
field wkpkg_wkp_sum_idx                       type composite
wkpkg_cmd, wkpkg_pe, wkpkg_proj, wkpkg_task, wkpkg_no
field wkpkg_no_idx                            type composite
wkpkg_cmd, wkpkg_cat, wkpkg_no

data record length          175
number of records           4334
number of indexes           7
wkpkg_proj                  duplicates allowed
wkpkg_no                     duplicates allowed
wkpkg_pdip                   duplicates allowed
wkpkg_proj_idx              duplicates allowed
wkpkg_wkpsys_idx            duplicates not allowed
wkpkg_wkp_sum_idx           duplicates not allowed
wkpkg_no_idx                 duplicates not allowed

```

file wkpkgfund permission all

field wkpf_cmd type char 12
field wkpf_cat type char 3
field wkpf_subcat type char 4
field wkpf_pe type char 5
field wkpf_proj type char 4
field wkpf_task type char 9
field wkpf_wkpkg type char 7
field wkpf_yr type integer
field wkpf_guid type long
field wkpf_fund type long
field wkpf_unfund type long
field wkpf_wkpkg_idx type composite
wkpf_cmd, wkpf_cat, wkpf_wkpkg
field wkpf_idx type composite
wkpf_cmd, wkpf_cat, wkpf_wkpkg, wkpf_yr

data record length 58
number of records 15736
number of indexes 2
wkpf_wkpkg_idx duplicates allowed
wkpf_idx duplicates not allowed

file wkstmt permission all

field wks_cmd type char 12
field wks_cat type char 3
field wks_subcat type char 4
field wks_pe type char 5
field wks_proj type char 4
field wks_task type char 9
field wks_wkpkg type char 7
field wks_yr type integer
field wks_text type char 600
field wks_wkpkg_idx type composite
wks_cmd, wks_cat, wks_wkpkg
field wks_idx type composite
wks_cmd, wks_cat, wks_wkpkg, wks_yr

data record length 646
number of records 10045
number of indexes 2
wks_wkpkg_idx duplicates allowed
wks_idx duplicates not allowed

A.1.5 CSS/EMW/SOF MAMP Specific Files

These permanent files have been created at BRDEC to specifically support the submissions of the CSS/EMW/SOF MAMPs. They are not present at other MAMP sites. In the case of the "com" and "lrrdppri" files, the data contained in them may be absorbed into the AMC system (in which case the data will need to be transferred and the ACE reports revised). The other files deal with the shifting of deficiency numbers from BDP 85 to BDP 86, and the effect of potential realignment of LRRDAP priorities in POM 89. They will quickly become outdated.

```
file com                                permission all

field com_com_line                      type char 10
field com_acq_strat                     type char 600

data record length                      614
number of records                       15
number of indexes                       1
com_com_line                            duplicates allowed
```

```
file defx                               permission all

field defx_85                           type integer
field defx_86                           type integer
field defx_idx                           type composite
      defx_86, defx_85

data record length                      4
number of records                       356
number of indexes                       3
defx_85                                 duplicates allowed
defx_86                                 duplicates allowed
defx_idx                                 duplicates not allowed
```

```
file lrrdppri                           permission all

field lrrdppri_pdip                     type char 6
field lrrdppri_pri                      type float
field lrrdppri_title                    type char 20
field lrrdppri_tradoc_pro               type char 3
field lrrdppri_tradoc_ma                type char 3
field lrrdppri_da_ma                    type char 3

data record length                      43
number of records                       638
number of indexes                       1
lrrdppri_pdip                          duplicates not allowed
```



```

file pom89                                     permission all

field pom89_pdip                             type char 6
field pom89_title                            type char 20
field pom89_pri                              type float
field pom89_tradoc_pro                       type char 3
field pom89_tradoc_ma                        type char 3
field pom89_da_ma                            type char 3
field pom89_strat                            type char 600

data record length      643
number of records      627
number of indexes       1
pom89_pdip              duplicates not allowed

```

```

file pom89ssn                                 permission all

field pom89ssn_ssn                          type char 6
field pom89ssn_pdip                          type char 6
field pom89ssn_pom88                         type char 6
field pom89ssn_seq                           type float
field pom89ssn_sys_idx                       type composite
      pom89ssn_ssn, pom89ssn_pom88

data record length      22
number of records      1264
number of indexes       3
pom89ssn_ssn            duplicates allowed
pom89ssn_pdip           duplicates allowed
pom89ssn_sys_idx        duplicates not allowed

```

A.2 Temporary Data Files

A.2.1 RDA Funding Rollup Files

These files roll up the funding data from the RDA database into consolidated files for easier access by ACE programs. The funding data is normally presented in the RDA and MAMP systems as individual records for each workpackage and year. Thus a workpackage may have as many as eight records to describe its funding stream. When an ACE report conducts a joining operation on that workpackage, for example taking the workpackage description and joining it with its funding data, the ACE system creates a copy of the description for each of the eight funding years. This gets particularly wasteful when performing more complex joins of deficiencies to systems to workpackages to funding years. And naturally, the ACE system bogs down in managing the thousands of copies of duplicate data fields.

These files translate the individual funding records into a single consolidated record for each workpackage with all eight years represented. The flagr file allows for six flags for each workpackage. The shells "wkpfrroll" and "flagroll" automatically build these files.

```
file wkpfr                                permission all

field wkpfr_cmd                            type char 12
field wkpfr_cat                            type char 3
field wkpfr_wkpkg                          type char 7
field wkpfr_baseyr                         type integer
field wkpfr_g0                             type long
field wkpfr_f0                             type long
field wkpfr_u0                             type long
field wkpfr_g1                             type long
field wkpfr_f1                             type long
field wkpfr_u1                             type long
field wkpfr_g2                             type long
field wkpfr_f2                             type long
field wkpfr_u2                             type long
field wkpfr_g3                             type long
field wkpfr_f3                             type long
field wkpfr_u3                             type long
field wkpfr_g4                             type long
field wkpfr_f4                             type long
field wkpfr_u4                             type long
field wkpfr_g5                             type long
field wkpfr_f5                             type long
field wkpfr_u5                             type long
field wkpfr_g6                             type long
field wkpfr_f6                             type long
field wkpfr_u6                             type long
field wkpfr_g7                             type long
field wkpfr_f7                             type long
field wkpfr_u7                             type long
field wkpfr_wkpkg_idx                       type composite
      wkpfr_cmd, wkpfr_cat, wkpfr_wkpkg

data record length          124
number of records          3984
number of indexes          1
wkpfr_wkpkg_idx           duplicates not allowed
```

file flagr permission all

field flagr_cmd	type char 12
field flagr_cat	type char 3
field flagr_wkpkg	type char 7
field flagr_wkpkg_idx	type composite
flagr_cmd, flagr_cat, flagr_wkpkg	
field flagr1_n	type char 5
field flagr1_0	type long
field flagr1_1	type long
field flagr1_2	type long
field flagr1_3	type long
field flagr1_4	type long
field flagr1_5	type long
field flagr1_6	type long
field flagr1_7	type long
field flagr2_n	type char 5
field flagr2_0	type long
field flagr2_1	type long
field flagr2_2	type long
field flagr2_3	type long
field flagr2_4	type long
field flagr2_5	type long
field flagr2_6	type long
field flagr2_7	type long
field flagr3_n	type char 5
field flagr3_0	type long
field flagr3_1	type long
field flagr3_2	type long
field flagr3_3	type long
field flagr3_4	type long
field flagr3_5	type long
field flagr3_6	type long
field flagr3_7	type long
field flagr4_n	type char 5
field flagr4_0	type long
field flagr4_1	type long
field flagr4_2	type long
field flagr4_3	type long
field flagr4_4	type long
field flagr4_5	type long
field flagr4_6	type long
field flagr4_7	type long
field flagr5_n	type char 5
field flagr5_0	type long
field flagr5_1	type long
field flagr5_2	type long
field flagr5_3	type long
field flagr5_4	type long
field flagr5_5	type long
field flagr5_6	type long
field flagr5_7	type long
field flagr6_n	type char 5
field flagr6_0	type long
field flagr6_1	type long
field flagr6_2	type long
field flagr6_3	type long
field flagr6_4	type long
field flagr6_5	type long
field flagr6_6	type long
field flagr6_7	type long

data record length	244
number of records	651
number of indexes	1
flagr_wkpkg_idx	duplicates not allowed

A.2.2 Periodically Updated Temporary Files

These temporary files need to be updated frequently to ensure that the latest changes in the primary and control permanent files are reflected in the MAMP reports. The shell "priority" automatically updates these files. It takes approximately 45 minutes to run.

```
file asrr                                     permission all

field asrr_ssn                               type char 6
field asrr_ssn1                              type char 6
field asrr_ssn2                              type char 6
field asrr_ssn3                              type char 6
field asrr_ssn4                              type char 6
field asrr_ssn5                              type char 6
field asrr_ssn6                              type char 6
field asrr_ssn7                              type char 6
field asrr_ssn8                              type char 6
field asrr_ssn9                              type char 6
field asrr_ssn10                             type char 6
field asrr_ssn11                             type char 6
field asrr_ssn12                             type char 6
field asrr_ssn13                             type char 6
field asrr_ssn14                             type char 6
field asrr_ssn15                             type char 6

data record length      100
number of records      279
number of indexes       1
asrr_ssn                duplicates not allowed
```

```
file hipri                                     permission all

field hipri_ssn                               type char 6
field hipri_pdip                              type char 6
field hipri_pri                               type float
field hipri_ok                                type integer
field hipri_title                             type char 60
field hipri_sys_idx                           type composite
      hipri_ssn, hipri_pdip

data record length      78
number of records      702
number of indexes       3
hipri_ssn                duplicates not allowed
hipri_pdip               duplicates allowed
hipri_sys_idx            duplicates not allowed
```

```

file prior1                                     permission all

field prior1_ssn_no                            type char 6
field prior1_ndef                             type integer
field prior1_na                               type integer
field prior1_nb                               type integer
field prior1_nc                               type integer
field prior1_nd                               type integer
field prior1_ne                               type integer
field prior1_nx                               type integer
field prior1_b1                              type integer
field prior1_score                            type float

data record length                            30
number of records                            656
number of indexes                            1
prior1_ssn_no                               duplicates not allowed

```

```

file prior2                                     permission all

field prior2_cmd                              type char 12
field prior2_cat                              type char 3
field prior2_wkpkg                            type char 7
field prior2_wkpkg_idx                       type composite
      prior2_cmd, prior2_cat, prior2_wkpkg
field prior2_nsys                             type integer
field prior2_hisys                           type float
field prior2_losys                           type float
field prior2_score                           type float

data record length                            40
number of records                            1033
number of indexes                            2
prior2_cmd                                   duplicates allowed
prior2_wkpkg_idx                             duplicates not allowed

```

```

file unissn                                     permission all

field unissn_ssn                             type char 6

data record length                            6
number of records                            702
number of indexes                            1
unissn_ssn                                   duplicates not allowed

```

file ssndroll permission all

field ssnd_ssn	type char 6
field ssnd_def1	type integer
field ssnd_con1	type char 1
field ssnd_def2	type integer
field ssnd_con2	type char 1
field ssnd_def3	type integer
field ssnd_con3	type char 1
field ssnd_def4	type integer
field ssnd_con4	type char 1
field ssnd_def5	type integer
field ssnd_con5	type char 1
field ssnd_def6	type integer
field ssnd_con6	type char 1
field ssnd_def7	type integer
field ssnd_con7	type char 1
field ssnd_def8	type integer
field ssnd_con8	type char 1
field ssnd_ndef	type integer

data record length	36
number of records	656
number of indexes	1
ssnd_ssn	duplicates not allowed

file uniwkp permission all

field uniwkp_cmd	type char 12
field uniwkp_cat	type char 3
field uniwkp_subcat	type char 4
field uniwkp_pe	type char 5
field uniwkp_proj	type char 4
field uniwkp_task	type char 9
field uniwkp_wkpkg	type char 7
field uniwkp_ssn	type char 6
field uniwkp_srf	type integer
field uniwkp_pdip	type char 6
field uniwkp_pom89	type char 6
field uniwkp_seq	type float
field uniwkp_sys_idx	type composite
uniwkp_ssn, uniwkp_pdip	
field uniwkp_wkpkg_idx	type composite
uniwkp_cmd, uniwkp_cat, uniwkp_subcat, uniwkp_pe, uniwkp_proj, uniwkp_task, uniwkp_wkpkg	

data record length	68
number of records	778
number of indexes	5
uniwkp_ssn	duplicates allowed
uniwkp_pdip	duplicates allowed
uniwkp_pom89	duplicates allowed
uniwkp_sys_idx	duplicates allowed
uniwkp_wkpkg_idx	duplicates not allowed

A.2.3 ACE Report Processing Files

These temporary files are used in the creation of the more complex ACE reports. They contain intermediate data that is used in the preparation of specific reports and is otherwise meaningless. The file fundpro has the deficiency, commodity, or system funding data (separated by year, tech base or development category, and funded/unfunded quantities). The file idxpg contains the page numbers for each ssn discovered in the "sysdollars.out" file by the C program "readidx".

```
file fundpro                                permission all

field fp_defic                             type integer
field fp_commodity                         type char 10
field fp_ssn                               type char 6
field f0_tb                                type long
field f1_tb                                type long
field f2_tb                                type long
field f3_tb                                type long
field f4_tb                                type long
field f5_tb                                type long
field f6_tb                                type long
field f7_tb                                type long
field u0_tb                                type long
field u1_tb                                type long
field u2_tb                                type long
field u3_tb                                type long
field u4_tb                                type long
field u5_tb                                type long
field u6_tb                                type long
field u7_tb                                type long
field f0_dev                               type long
field f1_dev                               type long
field f2_dev                               type long
field f3_dev                               type long
field f4_dev                               type long
field f5_dev                               type long
field f6_dev                               type long
field f7_dev                               type long
field u0_dev                               type long
field u1_dev                               type long
field u2_dev                               type long
field u3_dev                               type long
field u4_dev                               type long
field u5_dev                               type long
field u6_dev                               type long
field u7_dev                               type long

data record length                          150
number of records                          6
number of indexes                          3
fp_defic                                   duplicates allowed
fp_commodity                               duplicates allowed
fp_ssn                                     duplicates allowed

file idxpg                                  permission all

field idxpg_ssn                             type char 6
field idxpg_no                              type integer

data record length                          12
number of records                          403
number of indexes                          1
idxpg_ssn                                  duplicates allowed
```

APPENDIX B

MISCELLANEOUS PROCEDURES

B.1 Shell Programs for Temporary File Reconstitution

The shell programs "priority", "wkpfroll", and "flagroll" automatically update the temporary files with the current MAMP system data. "Wkpfroll" and "flagroll" only need to be executed once for each update of the RDA derived permanent files, specifically the workpackage funding data file and the flag data file. "Priority" should be run more frequently, after major changes to the permanent files and before running production MAMP reports.

"Priority" updates seven of the temporary files. It received its name because at first it only created the system and workpackage BDP rating files (prior1 and prior2). Since then, it has been expanded to include the unique controlled ssn file (unissn), the highest priority PDIP file (hipri), the system to deficiency linkage rollup file (ssndroll), the associated ssn rollup file (assr), and the unique workpackage to system linkage file (uniwkp). "Priority" is a convenient and easy way to ensure that the temporary files have been rebuilt and do reflect the current data in the MAMP system.

Aug 4 09:19 1986 priority Page 1

```
cd /u/review2/db
echo "see priority.log for a log of this procedure"
```

```
dbstatus mamp <<X      >>priority.log
erase file priori
y
erase file prior2
y
erase file unissn
y
erase file hipri
y
erase file ssndroll
y
erase file asr
y
erase file uniwp
y
bye
%
```

```
mv /u/review2/schema/prior1.k .
mv /u/review2/schema/prior2.k .
mv /u/review2/schema/unissn.k .
mv /u/review2/schema/hipri.k .
mv /u/review2/schema/ssndroll.k .
mv /u/review2/schema/asr.k .
mv /u/review2/schema/uniwp.k .
dbbuild prior1.k      >> priority.log
dbbuild prior2.k      >> priority.log
dbbuild unissn.k      >> priority.log
dbbuild hipri.k       >> priority.log
dbbuild ssndroll.k    >> priority.log
dbbuild asr.k         >> priority.log
dbbuild uniwp.k       >> priority.log
mv prior1.k /u/review2/schema
mv prior2.k /u/review2/schema
mv unissn.k /u/review2/schema
mv hipri.k /u/review2/schema
mv ssndroll.k /u/review2/schema
mv asr.k /u/review2/schema
mv uniwp.k /u/review2/schema
```

```
echo "building the unissn file"
informer mamp <<X      >> priority.log
read into a unique scl_ssn;
add noprompt unissn unissn_ssn = a.scl_ssn;
bye
%
```

```
echo "building the prior1 file"
acego .. /neil/prior1 >> priority.log
dbstatus mamp <<X      >> priority.log
load ascii prior1 from "prior1.out"
bye
%
```

Aug 4 09:19 1986 priority Page 2

```
echo "building the prior2 file"
acego ../neil/prior2 >> priority.log
dbstatus mamp <<% >> priority.log
load ascii prior2 from "prior2.out"
bye
%

echo "building the hipri file"
acego ../neil/hipri >> priority.log
dbstatus mamp <<% >> priority.log
load ascii hipri from "hipri.out"
bye
%

echo "building the ssndroll file"
acego ../neil/ssndroll >> priority.log
dbstatus mamp <<% >> priority.log
load ascii ssndroll from "ssndroll.out"
bye
%

echo "building the assr file"
acego ../neil/joinassr >> priority.log
dbstatus mamp <<% >> priority.log
load ascii assr from "joinassr.out"
bye
%

echo "building the uniwkp file"
acego ../neil/tsw >> priority.log
dbstatus mamp <<% >> priority.log
load ascii uniwkp from "tsw.out"
bye
%

/bin/rm joinassr.out
/bin/rm ssndroll.out
/bin/rm prior1.out
/bin/rm prior2.out
/bin/rm hipri.out
/bin/rm tsw.out
echo "all done with priority shell"
```

Aug 7 11:19 1986 flagroll Page 1

cd /u/review2/db

```
dbstatus mamp <<% > flagroll.log
erase file flagr
y
bye
%
```

```
mv ../schema/flagr.k .
dbbuild flagr.k >> flagroll.log
mv flagr.k ../schema
```

```
acego ../neil/xferflag 1986 >> flagroll.log
```

```
dbstatus mamp <<% >> flagroll.log
load ascii flagr from "flagr.out"
bye
%
```

/bin/rm flagr.out

Aug 7 11:19 1986 wkpfroll Page 1

cd /u/review2/db

```
dbstatus mamp <<% > wkpfr.out
erase file wkpfr
y
bye
%
```

```
mv ../schema/wkpfr.k .
dbbuild wkpfr.k >> wkpfr.out
mv wkpfr.k ../schema
```

```
acego ../neil/wkpfroll 1986 >> wkpfr.out
```

```
dbstatus mamp <<% >> wkpfr.out
load ascii wkpfr from "wkpfr.out"
bye
%
```

/bin/rm wkpfr.out

B.2 Shell Programs for Complex Report Execution

Several of the MAMP reports are multi-stage reports because of their inherent complexity and the limitations of the ACE programming system to accept the very large intermediate files they might create as single-stage reports. The multi-stage report first executes a subordinate ACE report, generating some intermediate values of interest. Then, these values are automatically read into a temporary file which is used by a second ACE report. This second report produces the final output that becomes the printed document.

The Deficiency Summary (defroll), Deficiency Funding Profile (defpro), and the Commodity Funding Profile (compro) are the three reports with this complex multi-stage execution. The shells defroll, defpro, and defroll, in the "../shell" directory, are used to execute these reports. Each of them creates a temporary data set which is loaded into the file "fundpro", and then the main ACE report is run. The calling sequence for these shells is:

```
expr [first pg#] | ../shell/defroll [pg letter] [code] [name]
expr [first pg#] | ../shell/defpro [pg letter] [code] [name]
expr [first pg#] | ../shell/compro [pg letter] [code] [name] [commodity]
```

The explanation for these calling sequences is contained in the introduction to Appendix C.

```
Aug 7 10:24 1986 defroll Page 1

cd /u/review2/db

dbstatus mamp <<%
erase file fundpro
y
bye
%

mv ../schema/fundpro.k .
dbbuild fundpro.k
mv fundpro.k ../schema

acego ../neil/syspro1 ${2} ${3}

dbstatus mamp <<%
load ascii fundpro from "syspro1.out"
bye
%

/bin/rm syspro1.out

acego ../neil/defroll ${1} ${2} ${3}
```

Aug 7 10:44 1986 defpro Page 1

```
cd /u/review2/db
dbstatus mamp <<%
erase file fundpro
y
bye
%

mv ../schema/fundpro.k .
dbbuild fundpro.k
mv fundpro.k ../schema

acego ../neil/defpro1 ${2} ${3}

dbstatus mamp <<%
load ascii fundpro from "defpro1.out"
bye
%

/bin/rm defpro1.out

acego ../neil/defpro2 ${1} ${2} ${3}
```

Aug 7 10:37 1986 compro Page 1

```
cd /u/review2/db
dbstatus mamp <<%
erase file fundpro
y
bye
%

mv ../schema/fundpro.k .
dbbuild fundpro.k
mv fundpro.k ../schema

acego ../neil/compro1 ${2} ${3} "${4}"

dbstatus mamp <<%
load ascii fundpro from "compro1.out"
bye
%

/bin/rm compro1.out

acego ../neil/compro2 ${1} ${2} ${3} "${4}"
```

B.3 Shell Programs for MAMP Production

Shell programs are also used for the rapid and convenient execution of the MAMP production runs. During the MAMP production period, many reports are prepared. Usually this tedious sequence of reports for one mission area is the same for all mission areas. Attention must be paid to the report's page numberings since they might depend on the length of previous reports. In addition, sometimes the execution of the many reports takes a long time and must be performed overnight. The shells described in this paragraph take away the drudgery of producing the MAMP reports by automatically stepping through the correct sequence of operating system calls needed to prepare the MAMP.

The following shell programs, found in the "../shell" directory, illustrate the use of shells to produce the MAMP. These are tailored for each mission area, and can be expected to change with each MAMP production run as the specific reports desired will change.

"Deficiency.css" produces the deficiency volume of the CSS MAMP. It creates the Deficiency Index, the Commodity Index, the Project Index, the Deficiency Summary, the Commodity Summary, the Project Summary, the PDIP to System Cross Reference, the BDP System Rankings, the BDP Workpackage Rankings, the LRRDAP Development System Priorities, the LRRDAP Workpackage Priorities, and the LRRDAP Procurement System Priorities. Each report is correctly paginated and controlled for printing. At the conclusion of the shell, the reports are moved to the "../output/cssreports" directory for storage. To verify the performance of the ACE reports, the file "defic.log" is available in the "../db" directory, which provides the ACE report output.

The simpler "system.css" and "wkpkg.css" similarly produce the system and workpackage volumes of the CSS MAMP. Again the reports are placed in the "../output/cssreports" directory, and the log files are created to verify the performance of the reports.

Aug 7 10:47 1986 deficiency.css Page 1

```
echo "SEE DEFIC.LOG FOR A LOG OF THIS PROCEDURE"  
cd /u/review2/db
```

```
expr 1 : acego ../neil/defindex a 1 CSS N > defic.log  
echo "FILE DEFINDEX1.OUT IS READY TO PRINT"  
expr 1 : acego ../neil/comindex b 1 CSS N >> defic.log  
echo "FILE COMINDEX1.OUT IS READY TO PRINT"  
expr 1 : acego ../neil/projindex c 1 CSS >> defic.log  
echo "FILE PROJINDEX.OUT IS READY TO PRINT"
```

```
expr 1 : ../shell/defroll Z 1 CSS >> defic.log  
echo "FILE DEFROLL1.OUT IS READY TO PRINT"
```

```
tail -10c defroll.out ; podd : acego ../neil/commod Z 1 CSS >> defic.log  
echo "FILE COMMODO.OUT IS READY TO PRINT"
```

```
tail -10c commod.out ; podd : acego ../neil/projwkp Z 1 CSS >> defic.log  
echo "FILE PROJWKP.OUT IS READY TO PRINT"
```

```
expr 1 : acego ../neil/pdipsys A 1 CSS >> defic.log  
echo "FILE PDIPSYS.OUT IS READY TO PRINT"  
expr 1 : acego ../neil/priorsys B 1 CSS >> defic.log  
echo "FILE PRIORSYS.OUT IS READY TO PRINT"  
expr 1 : acego ../neil/priorwp C 1 CSS >> defic.log  
echo "FILE PRIORWP.OUT IS READY TO PRINT"  
expr 1 : acego ../neil/lrrdpsys D 1 CSS C >> defic.log  
echo "FILE LRRDPSYS.OUT IS READY TO PRINT"  
expr 1 : acego ../neil/lrrdpwp E 1 CSS C >> defic.log  
echo "FILE LRRDPWP.OUT IS READY TO PRINT"  
expr 1 : acego ../neil/procpri F 1 CSS C >> defic.log  
echo "FILE PROCPRI.OUT IS READY TO PRINT"
```

```
mv defindex.out ../output/cssreports  
mv comindex.out ../output/cssreports  
mv projindex.out ../output/cssreports  
mv defroll.out ../output/cssreports  
mv commod.out ../output/cssreports  
mv projwkp.out ../output/cssreports  
mv pdipsys.out ../output/cssreports  
mv priorsys.out ../output/cssreports  
mv priorwp.out ../output/cssreports  
mv lrrdpsys.out ../output/cssreports  
mv lrrdpwp.out ../output/cssreports  
mv procpri.out ../output/cssreports  
mv defic.log ../output/cssreports  
echo "all done with deficiency volume"
```


Aug 7 10:55 1986 system.css Page 1

cd /u/review2/db

```
expr 1 : acego ../neil/sysres Z 1 CSS "*" > system.log
echo "FILE SYSRES.OUT IS READY TO PRINT"
```

```
tail -10c sysres.out : podd : acego ../neil/sydollars Z 1 CSS "*" >> system.log
echo "FILE SYSDOLLARS.OUT IS READY TO PRINT"
```

```
pidx < sydollars.out > pidx.out
informer mamp <<% >> system.log
delete noprompt idxpg;
bye
%
dbstatus mamp <<% >> system.log
load ascii idxpg from "pidx.out"
bye
%
/bin/rm pidx.out
```

```
expr 1 : acego ../neil/sysindex a 1 CSS >> system.log
echo "FILE SYSINDEX.OUT IS READY TO PRINT"
expr 1 : acego ../neil/sysindex1 b 1 CSS >> system.log
echo "FILE SYSINDEX1.OUT IS READY TO PRINT"
```

```
mv sysindex.out ../output/cssreports
mv sysindex1.out ../output/cssreports
mv sysres.out ../output/cssreports
mv sydollars.out ../output/cssreports
mv system.log ../output/cssreports
echo "all done with system volume"
```

Jun 13 17:04 1986 wkpkg.css Page 1

cd /u/review2/db

```
expr 1 : acego ../neil/wrkapdx Z 1 CSS "*" > wkpkg.log
echo "FILE WRKAPDX.OUT IS READY TO PRINT"
```

```
expr 1 : acego ../neil/wkpindex a 1 CSS "*" >> wkpkg.log
echo "FILE WKPINDEX.OUT IS READY TO PRINT"
```

```
mv wkpindex.out ../output/cssreports
mv wrkapdx.out ../output/cssreports
mv wkpkg.log ../output/cssreports
echo "all done with workpackage volume"
```

B.4 Shell Programs for Mass Actions to the Data Files

Occasionally, the user will need to make some mass changes to the database. These changes typically revolve around the deletion of an NSI or the change of an NSI to an SSN. Because the SSN is present in many different files, it can be extremely tedious to make all these changes and it is quite likely that one or more of the changes will not be made.

The shell programs presented below are used to rapidly change data records from one SSN to another, to delete all records with a particular SSN, or to print all data pertinent to a particular SSN. The correct calling sequence for these programs, found in the "../shell" directory, is:

```
updatessn [old ssn] [new ssn]
```

```
deletessn [ssn]
```

```
printssn [ssn]
```

Note that updatessn may have trouble in replacing the old ssn with the new ssn if corresponding records already exist for the new ssn. Then, the duplicate index value might cause Informix to not perform that update completely. The user might want to execute printssn for the new ssn first to see what data might already be present for the new ssn.

Aug 7 11:02 1986 updatessn Page 1

```
informer manp <<%
```

```
update noprompt assn assn_ssn = "%(2)" where assn_ssn = "%(1)";
update noprompt assn assn_assoc_ssn = "%(2)" where assn_assoc_ssn = "%(1)";
update noprompt lrrdp lrrdp_ssn = "%(2)" where lrrdp_ssn = "%(1)";
update noprompt ssn ssn_ssn = "%(2)" where ssn_ssn = "%(1)";
update noprompt ssnctl scl_ssn = "%(2)" where scl_ssn = "%(1)";
update noprompt ssndef ssndef_ssn = "%(2)" where ssndef_ssn = "%(1)";
update noprompt wkpsys wkpsys_ssn = "%(2)" where wkpsys_ssn = "%(1)";
```

```
update noprompt hipri hipri_ssn = "%(2)" where hipri_ssn = "%(1)";
update noprompt asrr asrr_ssn = "%(2)" where asrr_ssn = "%(1)";
update noprompt ssndroll ssnd_ssn = "%(2)" where ssnd_ssn = "%(1)";
update noprompt priori priori_ssn_no = "%(2)" where priori_ssn_no = "%(1)";
update noprompt unissn unissn_ssn = "%(2)" where unissn_ssn = "%(1)";
update noprompt pom89ssn pom89ssn_ssn = "%(2)" where pom89ssn_ssn = "%(1)";
update noprompt uniwkp uniwkp_ssn = "%(2)" where uniwkp_ssn = "%(1)";
update noprompt idxpg idxpg_ssn = "%(2)" where idxpg_ssn = "%(1)";
```

```
bye
%
```

Aug 7 11:03 1986 deletessn Page 1

informer mamp <<%

```
delete noprompt assn where assn_ssn = "%(1)";
delete noprompt assn where assn_assoc_ssn = "%(1)";
delete noprompt lrrdp where lrrdp_ssn = "%(1)";
delete noprompt ssn where ssn_ssn = "%(1)";
delete noprompt ssnctl where scl_ssn = "%(1)";
delete noprompt ssndef where ssndef_ssn = "%(1)";
delete noprompt wkpsys where wkpsys_ssn = "%(1)";

delete noprompt hipri where hipri_ssn = "%(1)";
delete noprompt asr where asr_ssn = "%(1)";
delete noprompt ssndroll where ssnd_ssn = "%(1)";
delete noprompt priori where priori_ssn_no = "%(1)";
delete noprompt unissn where unissn_ssn = "%(1)";
delete noprompt pom89ssn where pom89ssn_ssn = "%(1)";
delete noprompt uniwkp where uniwkp_ssn = "%(1)";
delete noprompt idxpg where idxpg_ssn = "%(1)";
```

bye
%

Aug 7 11:14 1986 printssn Page 1

informer -q mamp <<% > sysdata.out

```
print assn where assn_ssn = "%(1)";
print assn where assn_assoc_ssn = "%(1)";
print lrrdp where lrrdp_ssn = "%(1)";
print ssn where ssn_ssn = "%(1)";
print ssnctl where scl_ssn = "%(1)";
print ssndef where ssndef_ssn = "%(1)";
print wkpsys where wkpsys_ssn = "%(1)";

print hipri where hipri_ssn = "%(1)";
print asr where asr_ssn = "%(1)";
print ssndroll where ssnd_ssn = "%(1)";
print priori where priori_ssn_no = "%(1)";
print unissn where unissn_ssn = "%(1)";
print pom89ssn where pom89ssn_ssn = "%(1)";
print uniwkp where uniwkp_ssn = "%(1)";
print idxpg where idxpg_ssn = "%(1)";
```

bye
%
echo "system data is at sysdata.out"

B.5 Utility Programs

The following programs, written in the C language, are useful in managing the correct pagination of the MAMP volumes. Examples of the usage of these programs can be seen in the MAMP production shells described in B.3 above.

- (1) Pageread.c (compiled to "pnext")

This program gets a number from input, increments it, and writes it to output. It is used to identify the correct first page number for a printed report. Its usage would be:

```
tail -10c oldreport.out | pnext | acego newreport
```

- (2) Pageodd.c (compiled to "podd")

This program is similar to pnext, except that it finds the next odd number after the one input and writes that to output. This would be the correct page number if the new report must begin on a facing (odd) page. Its usage would be:

```
tail -10c oldreport.out | podd | acego newreport
```

- (3) Readidx.c (compiled to "pidx")

This program reads the ssn and page data off of the file "sysdollars.out" (which must be directed to input), inserts the proper delimiter characters, and writes the pagination data to output. The resulting data is loaded into the temporary database file "idxpg" for use in the two System Index Reports. This saves hours of manual work in searching for the data and entry into the indexes. A sample of the proper usage for this program is found in the shell "system.css", but is reproduced below:

```
pidx < sysdollars.out > pidx.out
informer mamp <<%
delete noprompt idxpg;
bye
%
dbstatus mamp <<%
load ascii idxpg from "pidx.out"
bye
%
```

Jun 13 13:12 1986 pageread.c Page 1

```
main()
{
    int i;
    scanf ("%d",&i);
    printf ("%d\n",i+1);
}
```

Jun 13 13:12 1986 pageodd.c Page 1

```
main()
{
    int i;
    scanf ("%d",&i);
    if (++i%2 == 0) i++;
    printf ("%d\n",i);
}
```

Jun 13 10:37 1986 readidx.c Page 1

```
# include <stdio.h>
main()
{
    char junk1;
    char junk2[28];
    char ssn[8];
    int i,j,page;

    j = 1;
    while (j != 0)
    {
        for (i = 1; i <= 6; i++)
        {
            while ((junk1 = getchar()) != '\n' && junk1 != EOF);
            if (junk1 == EOF) j = 0;
        }
        if (j == 1)
        {
            scanf ("%s %s", junk2, ssn);
            for (i = 1; i <= 56; i++)
            {
                while ((junk1 = getchar()) != '\n' && junk1 != EOF);
                if (junk1 == EOF) j = 0;
            }
            if (j == 1)
            {
                for (i = 1; i <= 61; i++) junk1 = getchar();
                scanf ("%d",&page);
                printf ("%s:%d:\n", ssn, page);
                for (i = 1; i <= 4; i++)
                {
                    while ((junk1 = getchar()) != '\n' && junk1 != EOF);
                    if (junk1 == EOF) j = 0;
                }
            }
        }
    }
}
```

B.6 Perform Screens

Two simple perform screens have been developed to assist data entry personnel in reviewing and updating the permanent MAMP data files. These perform screens are located in the "../db" directory and may be executed by using: "perform system", or "perform wkpsys" from that directory.

These screens are not as thorough or as comprehensive as the AMC provided ones. Instead, they provide only the minimum essential capability to locate, enter, and update the data present on the System Summary Report which has been used as the major format for quality control and data entry.

database manp

screen
(

```
System : [ssn ]
Title : [title ]
PDIP : [pdip ] DA Mission Area : [da ] TRADOC Mission Area : [tdc]
AMCMSC : [amcmsc ] AMC Manager : [cmd ] TRADOC Proponent : [pro]
Acquisition Type : [acq] [typ ] Cross Functional Area : [xfunc ]
Commodity : [com ] Req Doc : [rd1 ] [rd2 ] [rd3 ]
```

```
Description : Class : [c]
[d1 ]
[d2 ]
[d3 ]
[d4 ]
[d5 ]
[d6 ]
[d7 ]
[d8 ]
```

```
Deficiency : [def ] Contribution Value : [v] Associated SSN : [assoc ]
Control Code : [code] Control Name : [name ]
}
end
```

attributes

```
ssn = ssn_ssn = lrrdp_ssn = ssn_def_ssn = asssn_ssn = scl_ssn,
upshift, required, autonext,
comments = "Enter system SSN or NSI.";
title = lrrdp_title, upshift, autonext;
pdip = lrrdp_pdip, upshift, autonext;
da = ssn_da_ma, upshift, autonext;
tdc = ssn_tradoc_ma, upshift, autonext;
amcmsc = ssn_amc_msc, upshift, autonext,
include = (AMCHG, AMCCOM, AVSCOM, CECOM, LABCOM, MICOM, TACOM, TROSCOM, SDC),
comments = "Choices are: AMCHG, AMCCOM, AVSCOM, CECOM, LABCOM, MICOM, TACOM, TROSCOM, SDC";
cmd = ssn_amc_mgr, upshift, autonext;
pro = ssn_tradoc_pro, upshift, autonext;
acq = *acq_code = ssn_acq_code, autonext;
typ = acq_stt, autonext;
xfunc = ssn_xfuncarea, upshift, autonext;
com = ssn_com_line, upshift, autonext;
rd1 = ssn_req_doc1, upshift, autonext;
rd2 = ssn_req_doc2, upshift, autonext;
rd3 = ssn_req_doc3, upshift, autonext;

c = ssn_class, upshift, autonext;
d1 = ssn_desc[1,75], upshift, autonext;
d2 = ssn_desc[76,150], upshift, autonext;
d3 = ssn_desc[151,225], upshift, autonext;
d4 = ssn_desc[226,300], upshift, autonext;
d5 = ssn_desc[301,375], upshift, autonext;
d6 = ssn_desc[376,450], upshift, autonext;
```

Jul 3 13:50 1986 system.scr Page 2

```
d7 = ssn_desc[451,525], upshift, autonext;  
d8 = ssn_desc[526,600], upshift, autonext;
```

```
def = ssndef_def, autonext;  
v = ssndef_cont_value, upshift, autonext;  
assoc = assn_assoc_ssn, upshift, autonext;  
code = scl_code, autonext;  
name = scl_name, upshift, autonext;
```

instructions

```
ssn master of lrrdp;  
ssn master of ssndef;  
ssn master of assn;  
ssn master of ssnctl;
```

end

Jul 11 13:56 1986 wkpsys.scr Page 1

database manp

```
screen  
(
```

```
Command : [cmd      ]          Category : [cat][scat]  
Program Element : [pe  ]      Project : [proj]      Task      : [task    ]  
Workpackage : [wkpkg  ]
```

```
System : [ssn   ]          Criticality Factor : [w]
```

```
)  
end
```

attributes

```
cmd = wkpkg_cmd = wkpsys_cmd, required, upshift, autonext;  
cat = wkpkg_cat = wkpsys_cat, required, upshift, autonext,  
      include = (6.1, 6.2, 6.3);  
scat = wkpkg_subcat = wkpsys_subcat, required, upshift, autonext,  
      include = (6.1, 6.2, '6.3A', '6.3B', 6.4, 6.5, 6.7);  
pe = wkpkg_pe = wkpsys_pe, required, upshift, autonext;  
proj = wkpkg_proj = wkpsys_proj, required, upshift, autonext;  
task = wkpkg_task = wkpsys_task, required, upshift, autonext;  
wkpkg = wkpkg_no = wkpsys_wkpkg, upshift, required, autonext,  
      comments = "Enter wkpkg number.";
```

```
ssn = wkpsys_ssn, upshift, required, autonext,  
      comments = "Enter system SSN or NSI.";  
w = wkpsys_srf, upshift, required, autonext,  
      comments = "Enter wkpkg's criticality factor.";
```

instructions

```
composite join wkpkg_wkpsys_idx = wkpsys_wkpkg_idx;
```

```
wkpkg master of wkpsys;
```

end

APPENDIX C

SAMPLE OUTPUTS OF
CURRENTLY AVAILABLE MAMP REPORTS

The CSS/EMW/SOF MAMP system consists of over 50 ACE programs, most of which produce printed and formatted output. This appendix presents sample outputs of each report, along with a brief description of the report content and the correct method for initiating the running of the report.

The parameters used in each report are described on the following table. The user would substitute appropriate data for these parameters, to cause the program to produce the correct output. Reports which require the running of more than one ACE program are separately described in Appendix B, paragraph 2.

A standard procedure has been followed in the use of parameters in these reports. The first parameter is always the starting page number. It is provided as the UNIX standard input to the ACE program by piping it from the standard UNIX function 'expr'. It can also be provided by piping the output from the special C programs 'podd' or 'pnext' (see Appendix B, paragraph 5), or if the program is not run in the background then the user can type the number in directly from the keyboard when he is prompted for the starting page number.

The formal parameters for the ACE reports always begin with the sequence '[pg letter] [code] [name]'. The page letter to be used is substituted for the parameter '[pg letter]'. It allows the user to easily specify the page lettering to use (or to omit with a 'Z') that is consistent with the position of this particular report in the MAMP document that is being prepared. The parameters '[code]' and '[name]' are the control file filters that are applied in keeping with the AMC database structure. These would be used to select the MAMP report for data relevant to the CSS or EMW or SOF or BELVOIR mission areas.

The remaining parameters are included to add control and flexibility for each individual report. In most cases they allow the ACE report to be tailored from a lengthy report for the entire mission area to a much smaller report that specifically addresses the data of interest.

EXPLANATIONS OF PARAMETERS USED IN THE ACE REPORTS:

- [first pg#] an integer that describes the first page number to be printed on the report. Expr puts the page number into standard input so the report can be run in the background. In the foreground the user will be prompted to enter the page number from the terminal. See also the podd and pnext C programs.
- [pg letter] a single character that states the desired page letter to be printed along with the page number. A "Z" will cause the report to omit the page letter altogether.
- [code] an integer that references the code field in the mamp control files.
1 = TRADOC Mission Area
2 = DA Mission Area
3 = MSC
4 = Flag
- [command] a character string of the command name or "*" to indicate all commands.
- [name] a character string that designates the specific name entry for the code that is entered in the mamp control files.
- [class] a single character "U" or "C" that switches some reports between an unclassified and a confidential format.
- [base yr] an integer that indicates which year is designated by year 0 in the workpackage funding data.
- [ssn] a character string that designates a specific ssn or nsi to search on, or "*" to indicate all ssn's.
- [commodity] a character string of the commodity name, or "*" to indicate all commodity names. Use " " to contain commodity names with a space.
- [pro] a single character "Y" or "N" that switches the deficiency and commodity index reports between single column or double column entries for page numbers. Used if generating profiles.
- [proj] a character string that designates a specific RDTE project or an "*" to indicate all projects.
- [pdip] a character string to designate a specific pdip/increment or an "*" to indicate all pdips.

basetc

This report produces a streamlined summary of type classified systems. It used to also show base case systems, and can be reinstated quickly.

expr [first pg#] ; acego ../neil/basetc [pg letter] [code] [name] [command]

***** U N C L A S S I F I E D *****

STREAMLINED SUMMARY OF TYPE CLASSIFIED SYSTEMS FOR EMW

SSN	TITLE/DEFICIENCIES	DA/TDC MA	PROCUREMENT SCHEDULE																
			86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	
D00700	SEMITRAILER, LOW BED, 40T M870 (C/S) 41-D 306-D																		
E38801	MINE, AT HVY. M15 94-C																		
E38900	MINE, AP M16 94-C																		
E39100	MINE, AP M14 94-C																		
E39300	MINE, AP, DIRECTIONAL M18A1 94-C																		
E39900	ROCKET, SURFACE LAUNCHED UNIT FUEL AIR EXP-SL 29-D 86-D																		
E39800	MINE, AT HVY M21 94-C																		
E39900	MINE, AT M24 94-C																		
E40400	MINE, AT MK M19 94-C																		
E40601	GROUND IMP MINE SCAT SYS AP M74(MYP) 94-C																		
E43700	MINE, ANTIPERSONNEL, PRACTICE M68 94-C																		
E50300	CH DML BLK PENT 2LB M11B 29-C 94-C																		
E50700	DEMO KIT BANGALORE TORPEDO 94-C																		
E51000	CHG DEMO BLK TNT 1 LB 29-C 94-C																		
E51300	CHARGE, DEMO, 40LB, CRATERING 94-C																		
E52500	CAP, BLASTING, M6 SFC ELEC 94-D																		
E52600	CAP, BLASTING SPEC NON-ELECT M7 94-D																		
E53100	CHARGE, DEMO SHAPED M2A4 SERIES 15LB 94-C																		
E53200	CHARGE DEMO SHAPED 40LB M3 94-C																		
E54100	CORDE, DETONATING, WTRPRF TYPE 1 CLASS E 94-D																		

Legend: ***** - Funded Procurement
----- - Unfunded Procurement

***** U N C L A S S I F I E D *****

bdpalignmt

This report produces the BDP / LRRDAP alignment report, showing all systems and pdips and projects contributing to a specific bdp deficiency. This is the report that Randy Fowler requested.

expr [first pg#] ; acego ../neil/bdpalignmt [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

LRRDP/BDP 1986 ALIGNMENT FOR CSS

BDP DEFICIENCY	1986	13	1985	16	NSI/SSN/PE PROJ	PROGRAM	PROPONENT	CONT VALUE
					2D0001	FUZE, ARTILLERY MULTI-OPTION XM773	FAS	C
					2D0016	MODULAR CHARGE	FAS	A
					2D0017	UNI-CHARGE	FAS	A
					2D0019	PROPELLANT CHARGE XM-224	FAS	B
					9D0001	AMMUNITION PACKAGING TECHNOLOGY		B
456A-01					X00900	65 TON CRANE	MMS	B
					X50000	TRUCK, FORK LIFT, CBD, PT. 4000 LB.	MMS	B
456S-01					5L0003	AMMUNITION LOGISTICS	MMS	A
					ARDEC 63640 D543 A1	BATTLEFIELD RESUPPLY		1
					ARDEC 63640 D543 A1	ROBOTIC AMMUNITION RESUPPLY VEHICLE (RARV)		2
					ARDEC 63640 D543 A1	PALLETIZED LOADING SYSTEM (PLS) LOGISTICS DEMO		2
					ARDEC 63640 D543 A1	EXPLOSIVE ORDNANCE DISPOSAL PROOF OF PRINCIPLE		2
					ARDEC 63640 D543 A1	MUNITIONS STORAGE EVALUATION (QUICKLOAD)		2
					ARDEC 63640 D543 A1	UNIT BASIC LOAD-UPLOAD EQUIPMENT		2
					ARDEC 63640 D543 A1	UNIVERSAL TANK AMMUNITION PACKAGING (GALS)		2
					ARDEC 63640 D543 A1	GENERIC LP RESUPPLY VEHICLE		2
					ARDEC 63640 D543 A1	LOGISTICS C4 SYSTEM DEMONSTRATIONS		3
					ARDEC 63640 D543 A1	FUTURE HMITZER AMMUNITION LOGISTICS		2
					ARDEC 63640 D543 A1	AVIATION RESUPPLY LOGISTICS		2
					ARDEC 63640 D543 A1	AMMUNITION CONTAINER EVALUATION		2
					ARDEC 63640 D543 A1	155MM FIELD ARTILLERY AMMUNITION (FAA) TRAILER DEM		2
					ARDEC 63640 D543 A1	EXPLOSIVE ORDNANCE DISPOSAL PROOF OF PRINCIPLE		2
6M7G-01					3T0:06	IMPROVED NBC MAINTENANCE TECHNOLOGY	DCS	C
					NATICK 63747 D669 BV	POL RESISTANT CHEM PROT TACTILE GLOVE		1
					NATICK 64713 DL40 OY	DL40-OY INDIVIDUAL SOLDIER MICROCLIMATE COOLING SYSTEM		1
					NATICK 64713 DL40 IZ	DL40-IZ POL RESISTANT CHEMICAL PROTECTIVE GLOVE		1
6M72-01					3T0120	FAMILY OF CONTAINERS (FLEX)	GMS	C
6M7Y-01					5M2019	AFV - AMMO RESUPPLY (REARM)	ARS	E
6R5B-01					D05700	TRAILER, HEAVY, EXPANDED MOBILITY	TCS	E
6R5B-04					D05700	TRAILER, HEAVY, EXPANDED MOBILITY	TCS	E

***** U N C L A S S I F I E D *****

bpcss/bpemw

This report produces an unclassified bill payers report of workpackages in reverse order of the BDP sequence number located in bpcss/bpemw in the mamp. The files are separate for each mission area so that the sequence numbers can be kept straight. Cumulative totals are displayed.

expr [first pg#] | acego ../neil/bpcss [pg letter] [command]

bpcssa/bpemwa

Q
∞

This report produces an unclassified list of workpackages in decision aid format. This is the reverse order of bpcss/bpemw.

expr [first pg#] | acego ../neil/bpcssa [pg letter] [command]

bpdata

This report produces the temporary file which is read into the mamp database file bpcss or bpemw, depending upon the mission area parameter.

acego ../neil/bpdata [base yr] [code] [name]

***** U N C L A S S I F I E D *****

CSS N TO 1 BDP WORKPACKAGE BILLPAYERS
WITH CUMULATIVE TOTALS

PE/PROJ/TASK/WRKPKG		TITLE	FY86	FY87	FY88	FY89	FY90	FY91	FY92	FY93
305.	CECOM	63748 D594 01 315 METRO AND CALIBRATION	0	900	500	600	600	700	2200	0
		CUM TOTAL								
304	PMFSE	64718 DL82 02 6027 AD FACILITY INTRUSION DETECTION SYSTEM, GRP III (UNFUND)	0	900	500	600	1493	700	2200	0
		CUM TOTAL								
303.	PMFSE	64718 DL82 02 6004 BASIC FACILITY INTRUSION DETECTION SYSTEM (UNFUND)	0	900	500	600	2093	700	2200	0
		CUM TOTAL								
301	NATICK	64717 D429 03 D429-03 MATS (S280) HARDENED SHELTER	0	900	500	600	2093	700	2200	0
		CUM TOTAL								
300.	NATICK	64713 DL40 96 DL40-96 GROUND CVC CLOTHING SYSTEM (FACE MASK)	0	900	500	600	2093	700	2200	0
		CUM TOTAL								
299.	NATICK	64713 DL40 82 DL40-82 BODY ARMOR FOR EXPLOSIVE ORDNANCE DISPOSAL UNITS	0	900	500	600	2093	700	2200	0
		CUM TOTAL								
297.	NATICK	64713 D548 42 D548-42 ENG DEV OF GLCM FOOD SVC SYSTEM	0	900	500	600	2093	700	2200	0
		CUM TOTAL								
296.	TACOM	64604 D659 01 TV100 COMMERCIAL VEHICLES FOR TACTICAL APPLICATION	1968	900	500	600	2093	700	2200	0
		CUM TOTAL								
295.	NATICK	64218 D279 13 D279-13 BASIC MILITARY FREE-FALL PARACHUTE	1968	900	500	600	2093	700	2200	0
		CUM TOTAL								
294.	AVSCOM	64304 DC32 22 RTL4077 IN-HOUSE ENGINEERING SUPPORT - GROUND SUPPORT EQUIPMENT	390	900	500	600	2093	700	2200	0
		CUM TOTAL								
293	NATICK	63747 D669 34 D669-34 AIRCREW CLOTHING SYSTEM, COLD WEATHER	2358	900	500	600	2093	700	2200	0
		CUM TOTAL								
292.	NATICK	63747 D610 10 D610-10 FOOD PACKET, ASSAULT	2358	900	500	600	2093	700	2200	0
		CUM TOTAL								
290	BELVOIR	63726 DG14 28 2128 DIVING AND SUPPLY EQUIPMENT FOR NBC ENVIRONMENTS	2358	900	500	850	867	700	2200	0
		CUM TOTAL								
288	PM-AUC	63726 D526 06 2201 ARAPAH0 HELICOPTER SUPPORT SYSTEM	2358	900	500	850	2960	700	2200	0
		CUM TOTAL								

***** U N C L A S S I F I E D *****

Wed May 28 1986 15:03:49

cksyscmd

This report produces a diagnostic system summary, and shows missing data elements with underlines.

acego .../qa/cksyscmd [code, [name] [command]

***** S A M P L E *****

TITLE: TUNNEL DETECTION SYSTEM (TUDS)

SSN: 3T0002

Associated SSN's:

Mission Area: CS /EMW
Commodity Line: TUD/FLD F
Cross Functional Area:
Acquisition Type: Development

AMCMSC: TROSCOM
AMC Manager: BELVOIR
TRADOC Proponent: ENS
Req. Document: (LDA)

DEFICIENCIES: 265- , 319-8

DESCRIPTION: A NEW DEVELOPMENTAL SYSTEM TO PROVIDE THE ARMY WITH SOPHISTICATED ELECTRONIC HARDWARE & TECHNIQUES TO DETECT, LOCATE & CO
UNTER THE ENEMY INTRUSION THREAT POSED BY BOTH ACTIVE & PASSIVE TUNNELS AT DEPTHS UP TO 300 METERS IN ANY WORLDWIDE GEOL
OGICAL ENVIRONMENT. THE FOLLOWING SUBSYSTEMS ARE BEING DEVELOPED: (1) SEISMIC DETECTION SUBSYS; (2)ELECTROMAGNETIC DETECTIO
N SUBSYS; (3) RESISTIVITY DETECTION SUBSYSTEM; (4)FIELD DATA ANALYSIS SUBSYSTEM & (5) BOREHOLE TV INSPECTION SUBSYS. EXPLO
RATORY DEVELOP MT OF OTHER PROMISING DETECTION METHODS WILL BE CONT'D & CONSIDERED FOR LATER INCORP INTO TUDS.

PROCUREMENT FUNDING DATA

YEAR	FUNDED
1986	0
1987	0
1988	0
1989	0
1990	0
1991	0
1992	0
1993	0
1994	0
1995	0
1996	0
1997	0
1998	0
1999	0
2000	0
2001	0

WORKPACKAGE DATA

PE	PROJ	TASK	SUB	WPKG	TRAN	COMMAND	LAB	PRI	TITLE
64717	DH01	41	6.4	4015	---	BELVOIR	CED	---	TUNNEL DETECTION
63726	D001	31	6.3B	4014	---	BELVOIR	CED	50	TUNNEL DETECTION
62716	AH70	01	6.2	8207	---	LABCOM	HEL	360	ARMY APPLICATION OF ROBOTICS

***** S A M P L E *****

ckwtpkg

This report looks for unlinked workpackages.

acego ../qa/ckwtpkg [code] [name]

CSS FUNDED WORKPACKAGES NOT LINKED TO SYSTEMS

PE	PROJ	TASK	SUB	WPKPG	COMMAND	LAB	TITLE
62705	AH94	09	6.2	5017	LABCOM	ETDL	CRYSTAL OSCILLATORS
62705	AH94	09	6.2	5065	LABCOM	EIDL	ACUSTO-OPTICAL SIGNAL PROCESSING - II
62705	AH94	09	6.2	5066	LABCOM	EIDL	ACOUSTIC SIGNAL PROCESSING & FREQUENCY SOURCES - II
62705	AH94	09	6.2	5067	LABCOM	ETDL	CRYSTAL OSCILLATORS - II
62705	AH94	12	6.2	5020	LABCOM	EIDL	HIGH ENERGY PULSERS
62705	AH94	12	6.2	5037	LABCOM	ETDL	TACTICAL MEGAWATT COMPONENTS
62705	AH94	12	6.2	5070	LABCOM	ETDL	HIGH ENERGY PULSERS - II
62705	AH94	16	6.2	5040	LABCOM	ETDL	VALUE ENGINEERING FOR ELECTRONIC COMPONENTS
62705	AH94	16	6.2	5030	LABCOM	ETDL	VALUE ENGINEERING FOR ELECTRONIC COMPONENTS
62705	AH94	16	6.2	5090	LABCOM	ETDL	VALUE ENGINEERING FOR ELECTRONIC COMPONENTS
62723	AH98	A	6.2	AH98-A3	NATICK	IPD	ANTHROPOMETRIC DATA BASE
62733	AH20	FE	6.2	2235	BELVOIR	LSD	ELECTRIC POWER
62733	AH20	FL	6.2	2230	BELVOIR	LSD	LOGISTICS RESEARCH
63104	D130	09	6.3A	5021	BELVOIR	LSD	
63218	D242	01	6.3A	D242-01	NATICK	AMED	DROP ZONE ASSEMBLY AIDS, ELECTROMAGNETIC
63218	D242	02	6.3A	D242-02	NATICK	AMED	ULTRA HIGH LEVEL CONTAINER AIRDROP SYSTEM
63218	D242	06	6.3A	D242-06	NATICK	AMED	ADVANCED LOW ALTITUDE PERSONNEL PARACHUTE SYSTEM
63218	D242	3X	6.3A	D242-X3	NATICK	AMED	SMALL UNIT AIRBORNE RECOVERY/DELIVERY SYSTEM
63225	D266	06	6.3B	D266 06	NATICK	AMED	BUNDLE DELIVERY SYSTEM FOR C141
63705	DK82	07	6.3B	6028	PMPSE	PMPSE	REAR AREA PHYSICAL SECURITY SYSTEM
63726	DK39	93	6.3B	2221	PM-PWL	LSD	STORAGE DEVICES - INCR
63726	DK41	23	6.3B	2081	PM-PWL	LSD	20 LITER PLASTIC FUEL CAN
63726	DK41	31	6.3B	2031	PM-PWL	LSD	AIRDROPPABLE REFUELING MODULE/COLLAPSIBLE DRUMS
63726	DK41	37	6.3B	2131	PM-PWL	LSD	FUEL SURVEILLANCE
63747	D610	04	6.3B	D610-04	NATICK	FED	THE ARMY COMBAT FIELD FEEDING SYSTEM
63747	D610	06	6.3B	D610-06	NATICK	FED	TRAY PACK MENU ITEMS
63747	D610	13	6.3B	D610-13	NATICK	FED	ONE BURNER SQUAD STOVE
63747	D669	2V	6.3B	D669-V2	NATICK	AMED	FIELD TOILET SYSTEM
63747	D669	46	6.3B	D669-46	NATICK	AMED	LEAFLET ROLLING MACHINE FOR SOF
63748	DJ28	02	6.3A	J280203	CECOM	TMDE	ATSE NETWORKING
64218	D279	6X	6.4	D279-X6	NATICK	AMED	SMALL UNIT AIRBORNE RECOV/DELIV SYSTEM
64604	D659	01	6.4	TV100	TACOM	PMHTV	FAMILY OF HEAVY TACTICAL VEHICLES (FHTV)
64713	DL40	4V	6.4	DL40-4V	NATICK	AMED	HIGH MOBILITY VEHICLE MAINTENANCE SHELTER
64713	DL40	82	6.4	DL40U5U	NATICK	AMED	MODULAR FAMILY OF TENTS/1990S
64713	DL40	82	6.4	DL40-82	NATICK	IPD	BODY ARMOR FOR EXPLOSIVE ORDNANCE DISPOSAL UNITS
64713	DL40	98	6.4	DL40-98	NATICK	IPD	COMBAT FOOTWEAR
64717	DH14	11	6.4	2016	BELVOIR	LSD	ROUGH TERRAIN CONTAINER TRANSPORTER (RTCT) - INCREMENT
65810	DE65	01	6.5	E650101	CECOM	PM TMDE	TEMOD EVALUATION (NDI) - IN-HOUSE SUPPORT
65810	DE65	01	6.5	E650102	CECOM	PM TMDE	TEMOD EVALUATION (NDI) - PROGRAM SUPPORT
65810	DE65	01	6.5	E650103	CECOM	PM TMDE	TEMOD EVALUATION (NDI) - FEASIBILITY
65810	DE65	4854367	6.5	4854367	TACOM	RC	COMMERCIAL TRAILERS FOR TACTICAL APPLICATION
65810	DE65	4854367	6.5	4854367	TACOM	TASL	M1 BULLDOZER KIT PROGRAM
65810	DE65	4856116	6.5	RC6116	TACOM	RT	EVALUATION OF RADIAL PLY TIRES
65810	DE65	4865023	6.5	4865023	TACOM	RG	MACI: TACTICAL TRUCK DRIVELINE COMPONENTS
65810	DE65	4865023	6.5	3022	BELVOIR	LSD	DIESELIZATION OF 400K BTUH HEATER
65810	DE65	EB63816	6.5	3018	BELVOIR	LSD	MACI-STD FAM AIR CONDS (OMNIBUS)
65810	DE65	EB63817	6.5	3019	BELVOIR	LSD	MACI-DOD STD FAM MFPGS (CEP PROG) (OMNIBUS)
65810	DE65	EB63827	6.5	2137	BELVOIR	LSD	MACI-CONTAINERIZATION
65810	DE65	EB63829	6.5	2139	BELVOIR	LSD	MACI-RAIL EQUIPMENT
65810	DE65	EB73818	6.5	2230	BELVOIR	LSD	ESCORT CAR
65810	DE65	TM4500	6.5	5041	TACOM	TMM	XM1067 TRUCK TRACTOR PROGRAM
65810	DE65	TM4500	6.5	5042	TACOM	TMM	HEAVY DUTY MEDIUM EQUIPMENT TRANSPORTATION PROGRAM.
65810	DE65	TM4500	6.5	5043	TACOM	TMM	HIGH MOBILITY TRAILER

comindex

This report produces an unclassified index of the commodity lines with two page entry columns, one for the commodity summary and one for the commodity profile. It is sorted by commodity line.

expr [first pg#] | acego ../neil/comindex [pg letter] [code] [name] [pro]

***** U N C L A S S I F I E D *****

CSS COMMODITY LINE INDEX IN ALPHABETICAL ORDER

COMMODITY LINE	SUMMARY PAGE	FUNDING PROFILE
AIR DROP	---	---
AMMO	---	---
AVN SPT EQ	---	---
CONST EQ	---	---
FD/FLD SVC	---	---
FUEL HDL	---	---
FUEL/LUBES	---	---
IND/PRO EQ	---	---
MAINT EQ	---	---
MARINE CFT	---	---
MAT HDL	---	---
MED SPT	---	---
MTL HDL	---	---
PHY SCTY	---	---
SHELTERS	---	---
SPT VEH	---	---
TAC VEH	---	---
TACGS	---	---
TMDE	---	---
WR PUR/DIS	---	---

***** U N C L A S S I F I E D *****

A- 1

Thu Jul 24 1986 13:55:00

commod

This report produces a unclassified summary of the commodities represented by systems in a designated mission area. System deficiencies and procurement funding are included. It is sorted by command and commodity line.

expr [first pg#] ; acego ../neil/commod [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

BRIDGING COMMODITY LINE SUMMARY FOR EMW

Acquisition Strategy:

PROCUREMENT SCHEDULE

BSN	TITLE/DEFICIENCIES	DA/TDC MA	PDIP	86	87	88	89	90	91	92	93	94	95	96	97	98	99
	***** TYPE CLASSIFIED *****																
M20000	LINK REINFORCING SET, MG BRIDGE	CS	/	EMW	6XES-02					*****							
	BELVOIR 41-D 133-D																
M21901	LIGHT TACTICAL RAFT	CS	/	EMW	6XES-01												
	BELVOIR 41-C 133-D																
M23600	RIBBON BRIDGE ERECTION BOAT	CS	/	EMW	6XES-01					*****							
	BELVOIR 133-B																
M23601	BRIDGE ERECTION BOAT, SD	CS	/	EMW	6XES-01												
	BELVOIR 133-C																
M24601	MAB, INT BAY BRIDGING	CS	/	EMW	6XES-01												
	BELVOIR 133-D																
M24700	M4700 FLOATING BRG/DRY SPAN	CS	/	EMW	6XES-01												
	BELVOIR 133-D																
M24801	MAB, RAMP BAY	CS	/	EMW	6XES-01												
	BELVOIR 133-E																
M25001	BRIDGE, BAILEY, M2	CS	/	EMW	-												
	BELVOIR 133-E																
M25301	CLASS 60 FLOAT BRG	CS	/	EMW	-												
	BELVOIR 133-E																
M29500	BRIDGE, FIXED, HIGHWAY, ALUM, MED GIRDER	CS	/	EMW	6XES-01												
	BELVOIR 133-C																
M29700	BRIDGE, ERECTION SET, MED GIRD	CS	/	EMW	6XES-03												
	BELVOIR 133-D																
MA7100	ITEMS LESS THAN \$2.0M(BRIDGING)	CS	/	EMW	6XES-01												
	BELVOIR 133-D																

3T0006	HEAVY DRY SUPPORT BRIDGE	CS	/	EMW	6XES-01												
	BELVOIR 80 133-B																
5M2014	AFV-IMPROVED HEAVY ASSAULT BRIDGE (IHAB)	CS	/	EMW	6M7Y-01												
	TACOM 41-C 65-A 86-B 131-C 133-B 307-B 313-D																

Legend: ***** - Funded Procurement

***** U N C L A S S I F I E D *****

compro1

This report produces a temporary file which is loaded into the database file fundpro. It accumulates the funding data for uniquely linked workpackages associated with each commodity line. The data is used in the commodity profile report compro2.

acego ../neil/compro1 [code] [name] [commodity]

compro2

This report produces an unclassified commodity profile, a vertical bar chart showing the total tech base, development, and procurement funding for a commodity line within a mission area.

expr [first pg#] | acego ../neil/compro2 [pg letter] [code] [name] [commodity]

comsyspdip

This report produces a listing of the commodity lines/systems/pdips which are appropriate to a particular mission area. It was requested by Bob Brown.

expr [first pg#] ; acego ../neil/comsyspdip [pg letter] [code] [name]

***** UNCLASSIFIED *****

BRIDGING COMMODITY/SYSTEM/PDIP SUMMARY FOR EMW

***** TYPE CLASSIFIED *****

M20000	LINK REINFORCING SET, MC BRIDGE		CS / EMW		BELVOIR		FY97	FY98	FY99	FY00	FY01
	PDIP	FY87	FY88	FY89	FY90	FY91					
6XE302	0	0	0	0	0	5000	0	0	0	0	0
6XE303	0	0	4800	0	5300	5000	0	0	0	0	0
Total:	0	0	4800	0	5300	10000	0	0	0	0	0
M21901	LIGHT TACTICAL RAFT		CS / EMW		BELVOIR		FY97	FY98	FY99	FY00	FY01
PDIP	FY87	FY88	FY89	FY90	FY91	FY92					
6XE301	0	0	0	0	0	0	0	0	0	0	0
M23600	RIBBON BRIDGE ERECTION BOAT		CS / EMW		BELVOIR		FY97	FY98	FY99	FY00	FY01
PDIP	FY87	FY88	FY89	FY90	FY91	FY92					
6XE301	19200	18800	8100	0	21500	0	20300	20300	20700	21700	21700
6XE303	0	0	20400	29200	8500	0	0	0	0	0	0
Total:	19200	18800	28500	29200	30000	0	20300	20300	20700	21700	21700
M23601	BRIDGE ERECTION BOAT, SD		CS / EMW		BELVOIR		FY97	FY98	FY99	FY00	FY01
PDIP	FY87	FY88	FY89	FY90	FY91	FY92					
6XE301	0	0	0	0	0	0	0	0	0	0	0
M24601	MAB, INT BAY BRIDGING		CS / EMW		BELVOIR		FY97	FY98	FY99	FY00	FY01
PDIP	FY87	FY88	FY89	FY90	FY91	FY92					
6XE301	0	0	0	0	0	0	0	0	0	0	0
M24700	M4700 FLOATING BRG/DRY SPAN		CS / EMW		BELVOIR		FY97	FY98	FY99	FY00	FY01
PDIP	FY87	FY88	FY89	FY90	FY91	FY92					
6XE301	0	0	0	0	0	0	0	0	0	0	0
M24801	MAB, RAMP BAY		CS / EMW		BELVOIR		FY97	FY98	FY99	FY00	FY01
PDIP	FY87	FY88	FY89	FY90	FY91	FY92					
6XE301	0	0	0	0	0	0	0	0	0	0	0
M25001	BRIDGE, BAILEY, M2		CS / EMW		BELVOIR		FY97	FY98	FY99	FY00	FY01
PDIP	FY87	FY88	FY89	FY90	FY91	FY92					
N/A	0	0	0	0	0	0	0	0	0	0	0
M25301	CLASS 60 FLOAT BRG		CS / EMW		BELVOIR		FY97	FY98	FY99	FY00	FY01
PDIP	FY87	FY88	FY89	FY90	FY91	FY92					
N/A	0	0	0	0	0	0	0	0	0	0	0
M29500	BRIDGE, FIXED, HIGHWAY, ALUM, MED GIRDER		CS / EMW		BELVOIR		FY97	FY98	FY99	FY00	FY01
PDIP	FY87	FY88	FY89	FY90	FY91	FY92					
6XE301	0	0	0	0	0	0	16000	16000	16000	0	0
6XE303	0	0	0	0	0	15900	0	0	0	0	0
Total:	0	0	0	0	0	15900	16000	16000	16000	0	0

***** UNCLASSIFIED *****

Tue Jul 29 1986 11:20:49

decsysbdp

This report produces an unclassified decision aid for systems in a mission area based on the BDP priority sequence. Developmental systems are listed in priority order, with base case systems first, and unfunded RDTE requirements for workpackages with a criticality value of 1, 2, or 3 are shown.

expr [first pg#] | acego ../neil/decsysbdp [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

EMW I TO N DEVELOPMENT SYSTEMS BDP RANKINGS
WITH SIGNIFICANT UNFUNDED WORKPACKAGES SHOWN

		UNFUNDED DEVELOPMENT SCHEDULE									
SSN	TITLE / UNFUNDED WORKPACKAGES	DA/TDC MA	FY86	FY87	FY88	FY89	FY90	FY91	FY92	FY93	
	BELVOIR 63606 D608 27 1340 UNFINANCED COUNTERMINE PROGRAM			(3999)	(15701)						
	BELVOIR 63606 D608 27 1341 UNFINANCED COUNTERMINE PROGRAM - INCREMENT			(1945)							
21.	3T0020 INTEGRATED POWER ENVIRONMENTAL CONTROL S	CS / EMW									
22.	6E0001 ADVANCED TACTICAL POWER SOURCES	CS / EMW									
23.	K42551 QUICK RESPONSE MULTI COLOR PRINTER	CS / EMW									
24.	3T0304 SURVIVABLE TACTICAL ARMY GENERATOR	CS / EMW									
25.	3T0304B LIGHT INFANTRY MINEFIELD BREACHING SYSTE	CS / EMW						(2450)			
	BELVOIR 63606 D608 07 1310 LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (LIMBS)-IM			(3999)	(15701)						
	BELVOIR 63606 D608 27 1340 UNFINANCED COUNTERMINE PROGRAM			(1945)							
	BELVOIR 63606 D608 27 1341 UNFINANCED COUNTERMINE PROGRAM - INCREMENT										
26.	6E0002 THERMOELECTRIC GENERATORS	CS / EMW			(1034)	(1134)	(1294)	(1741)	(1741)	(1741)	
	LABCOM 62705 AH94 11 5069 THERMOELECTRIC GENERATOR - IPECS - II										
27.	3T0300 GEN INTEG SYS FOR TACTICAL VEHICLES	CS / EMW									
28.	WC3112 COUNTEROBSTACLE VEHICLE (COV)	CS / EMW			(7000)	(13000)	(13000)	(8000)			
	BELVOIR 64C0V DCDV 01 1316 NEW PROPOSED PROGRAM FOR COV										
29.	3T0049 IMPROVED LINE CHARGE	CS / EMW									
30.	6T0007 COUNTERMINE CONCEPTS - BROAD BASED TECHN	CS / EMW									
	BELVOIR 65810 D125 01 1210 MARKET INVESTIGATION - COMBAT ENGINEERING DIRECTORAT			(700)	(850)	(1000)	(1104)	(1090)	(1215)	(1300)	
31.	R6300 THERMOELECTRIC GENERATOR 1.5KW	CS / EMW									
32.	3T0031 IMPROVED HAND HELD MINE DETECTOR	CS / EMW									
	BELVOIR 63606 D608 29 1300 IMPROVED HAND HELD MINE DETECTOR- INCREMENT			(3999)	(15701)						
	BELVOIR 63606 D608 27 1340 UNFINANCED COUNTERMINE PROGRAM										
	BELVOIR 63606 D608 27 1341 UNFINANCED COUNTERMINE PROGRAM - INCREMENT										
	BELVOIR 63606 D608 49 1302 ADVANCED TECHNOLOGY PORTABLE MINE DETECTOR - INCREME										
33.	3M0201 SEE VARIANT	CS / EMW									
	TACOM 62601 AH91 300 RC999 CORROSION OF FIBER REINFORCED COMPOSITE ARMOR				(141)	(212)	(351)	(523)	(550)		
34.	3T0112 FIELD ARMY MAPPING SYSTEMS (DTSS)	CS / EMW									

***** U N C L A S S I F I E D *****

Fri Jul 25 1986 11:36:18

decwpbdp

This report produces an unclassified decision aid for workpackages in a mission area based on BDP priority sequence, and shows unfunded requirements for those workpackages. (6.3, 6.4, and 6.7)

expr [first pg#] ; acego ../neil/decwpbdp [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

EMW 1 TO N WORKPACKAGE BDP RANKINGS
WITH UNFUNDED AMOUNTS SHOWN

PE/PROJ/TASK/WKPKG	TITLE	UNFUNDED DEVELOPMENT SCHEDULE									
		FY86	FY87	FY88	FY89	FY90	FY91	FY92	FY93		
1.	MINEFIELD RECONNAISSANCE AND DETECTOR SYSTEM (MIRADOR) ARDEC 63619 D606 27 01343										
2.	MINEFIELD RECONNAISSANCE AND DETECTOR SYSTEM (MIRADOR) ARDEC 64612 D415 10 01340										
3.	13/30/60 KW SIGNATURE SUPPRESSED GENERATOR SETS MEP 64714 D194 15 7008										
4.	MULTI-PURPOSE DETECTION SYSTEM AD (MIN) ARDEC 63619 D606 37 01351										
5.	COUNTEROBSTACLE VEHICLE BELVOIR 63606 D608 08 1006										
6.	HEAVY ASSAULT BRIDGE BELVOIR 63726 D601 01 1029										
7.	MINEFIELD RECONNAISSANCE AND DETECTOR SYSTEM (MIRADOR) ARDEC 63619 D606 27 91343										
8.	MULTI-PURPOSE DETECTION SYSTEM ARDEC 63619 D606 37 91351										
9.	WIDE AREA NEUTRALIZATION DEVICE (HAND) BELVOIR 63606 D608 16 1023										
10.	WIDE AREA NEUTRALIZATION DEVICE (HAND) (MIN) ARDEC 64612 D415 12 01355										
11.	BARRIER 21/ANTI-TANK OBSTACLE BELVOIR 63606 D608 01 1054										(1000) (1400) (2000)
12.	COUNTEROBSTACLE VEHICLE (CDV) ARDEC 63619 D601 31 91346										
13.	TACTICAL EXPLOSIVE SYSTEM (TEXS) ED (MIN) ARDEC 64612 D021 02 01325										
14.	AIRBORNE MINEFIELD DETECTION & RECONNAISSANCE SYSTEM BELVOIR 63606 D608 11 1007										
15.	MINEFIELD IMAGE RECOGNITION & CUEING STATIONS (MIRACS) BELVOIR 63606 D608 48 1315										
16.	AIRBORNE MINEFIELD DETECTION SYSTEM (AMIDARS) (MIN) ARDEC 64612 D415 09 01341										
17.	AIRBORNE MINEFIELD DETECTION SYSTEM (AMIDARS) (RISK) ARDEC 64612 D415 09 11341										(3999) (15701)
18.	CANE TIP BELVOIR 63606 D608 05 1005										
19.	TACTICAL EXPLOSIVE SYSTEM (TEXS) ARDEC 64612 D021 02 91325										
20.	UNFINANCED COUNTERMINE PROGRAM BELVOIR 63606 D608 27 1340										

***** U N C L A S S I F I E D *****

Fri Jul 25 1986 11:17:16

defindex

This report produces an unclassified index for the deficiencies in a mission area. Two page columns are printed, one for the deficiency rollup, and one for the deficiency profile, depending upon the value of pro.

expr [first pg#] : acego ../neil/defindex [pg letter] [code] [name] [pro]

***** U N C L A S S I F I E D *****

EMN DEFICIENCY INDEX IN NUMERICAL ORDER

DEFICIENCY	TITLE	SUMMARY PAGE	FUNDING PROFILE
29		---	---
41		---	---
86		---	---
94		---	---
95		---	---
106		---	---
131		---	---
132		---	---
133		---	---
145		---	---
230		---	---
237		---	---
306		---	---
307		---	---
310		---	---
313		---	---
319		---	---
320		---	---
322		---	---
325		---	---
326		---	---
327		---	---
331		---	---

***** U N C L A S S I F I E D *****

A- 1

Thu Jul 24 1986 14:26:02

defpro1

This report produces the temporary file which is read into the database file fundpro. It summarizes the rdte funding for a deficiency without duplicating the workpackages.

acego ../neil/defpro1 [code] [name]

defpro2

This produces an unclassified funding profile for the deficiencies within a mission area. It is just like the commodity profile.

expr [first pg#] ; acego ../neil/defpro2 [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

FISCAL SUMMARY - DEFICIENCY 29

	1986	1987	1988	1989	1990	1991	1992	1993	94	95	96	97	98	99	00	01
Tech Base	19057	54134	60189	106745	110644	110870	122927	140225								
XXX/XXX--	0	(7398)	(19201)	(10693)	(24943)	(31491)	(23763)	0								
Development	11570	28001	20874	16841	23280	23934	30197	27316								
(DDD/D--)	0	0	0	(7600)	(15000)	(15500)	(8400)	0								
Procurement	0	41800	39300	46800	83700	110500	135208	108208								
See Out Year Procurement																
					1994	1995	1996	1997	1998	1999	2000	2001				
					128308	124008	105108	96308	78208	33208	33208	33208				
					18000	13000	0	0	0	0	0	0				

***** U N C L A S S I F I E D *****

syspro1

This report produces the data file which is read into the temporary file fundpro. It rolls up the rdte funding for designated systems so that the deficiency rollup will have less work to do.

acego ../neil/syspro1 [code] [name]

defroll

This report produces the classified deficiency summary for all deficiencies in a mission area. System procurement and rdte funding is shown as a series of asterisks.

expr [first pg#] ! acego ../neil/defroll [pg letter] [code] [name]

AD-A172 652

CSS/EMM/SOF (COMBAT SERVICE SUPPORT/ENGINEERING AND
MINE WARFARE/SPECIAL (U) MCLEAN RESEARCH CENTER INC VA
G N ROMSTEDT SEP 86 DAAK70-84-D-0052

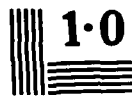
2/4

UNCLASSIFIED

F/G 9/2

NL





1.0

2.8

2.5

3.15

2.2

3.5

2.0

4.0

1.8

4.5



1.1



1.25



1.4



1.6

***** S A M P L E *****

TRADOC DEFICIENCY: 29 EMW

DESCRIPTION:

1. SCATERABLE MINE DETECTION SYSTEM
 ssn number: 6D0032 command: PM-MCD - CS / EMW
1. E DET SET, MINE, MET/NON-MET, AN/PRB-B
 ssn number: M15300 command: BELVOIR - CS / EMW
2. CH DML BLK PENT 21B M11B
 ssn number: E50500 command: AMCCOM - CSS / EMW
3. CHD DEMO BLK TNT 1 LB
 ssn number: E51000 command: ARDEC - CSS / EMW
4. LAUNCHER, SLUFAE
 ssn number: M96300 command: AMCCOM - CS / EMW
5. MINE CLEARING ROLLER
 ssn number: X00600 command: TACOM - CC / EMW
6. MARKING SYS, CLEAR LANE
 ssn number: X00700 command: PM-MCD - CS / EMW
7. ROCKET, SURFACE LAUNCHED UNIT FUEL AIR EXP-SLUFAE
 ssn number: E39500 command: AMCCOM - CS / EMW
8. TRANSPORTER, SLUFAE
 ssn number: 6A2525 command: AMCCOM - CS / EMW

Type Classified:

COM	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01
B																
B																
C																
C																
C																
C																
C																
D																
D																

***** S A M P L E *****

Thu Aug 7 1986 12:52:52

Legend:
 **** - System Production Funding
 ***** - Multi-def. System Production Funding
 +---- - Work Package RDTE Funding

defsys

This diagnostic report shows all systems linked to a deficiencies within a mission area. It shows if the systems are controlled or not.

expr [first pg#] i acego ../qa/defsys [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

EMW DEFICIENCIES WITH SYSTEM BDP RANKING

Item ID	Ranking	System Name
1M1101 *	C	MINE PLOW BLADE
3B6101	A	CONV TNG MINE
3T0046 *	D	ADVANCED MINE CLEARING SYS
3T0048 *	C	LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (LIMBS)
3T0049 *	C	IMPROVED LINE CHARGE
3T0051 *	D	IMPROVED HAND HELD MINE DETECTOR
3T0053 *	E	TACTICAL OBSTACLE MARKING SYSTEM (TOMS)
3T0056 *	D	REACTIVE FRAGMENT MINE NEUTRALIZATION SYSTEM (FRAGS)
3T0057 *	B	WIDE-AREA NEUTRALIZATION MINE DEVICE (HAND)
3T0058 *	D	MAGNETIC SIGNATURE SILENCING SYSTEM (MASSIS)
3T0307 *	C	ELECTRIC EXPLOSIVE DETONATION SYSTEM
3M2015 *	B	AFV-ENGINEER SQUAD (SAPPER)
3M2016 *	C	AFV-COMBAT EXCAVATOR (CEV)
3M2017 *	D	AFV - COMBAT EARTHMOVER (CEM)
3M2018 *	A	AFV-COMBAT MOBILITY VEHICLE (CMV)
3R1001	E	MOBILE LASER DEARMER (MOLAD)
3T0003 *	C	AIRBORNE MINEFIELD DETECTION & RECONN SYS--AMIDARS
3T0006 *	C	SCATTERMINE RADAR (SCMR)
3T0007 *	C	MULTIPURPOSE DETECTION SYSTEM
3T0028 *	C	CANETIP
3T0037 *	B	AUTOMATED COUNTERBARRIER SYSTEM
3T0038 *	C	VEHICLE HARDENING
3T0040 *	D	REAR AREA MINE CLEARING SYSTEM (RAMV)
3T0041 *	E	MAN MADE OBSTAC RPT & DUP SYS (MORDS)
3T0050		SITUATION SENSITIVE DETECTION TECHNOLOGY
3T0061		SITUATION SENSITIVE MOBILITY TECHNOLOGY
6D0026	B	ADVANCED MECHANICAL MINE CLEARING DEVICE
6D0032	B	SCATERABLE MINE DETECTION SYSTEM
6T0007 *	B	COUNTERMINE CONCEPTS - BROAD BASED TECHNICAL AREA
7B1801	A	ENG EX SYS ADV
734101	A	FASCAM EFFECT
7B4102	A	SAFE-NES
E39500 *	D	ROCKET, SURFACE LAUNCHED UNIT FUEL AIR EXP-SLUFAC
E50500 *	C	CH DML BLK PENT 2LB M11B
E51000 *	C	CHG DEMO BLK TNT 1 LB
E75100 *	B	LINE CHARGE M58A3 (MICLIC)
Q24000	B	ROBAT (M-60 MOD)
QAZ525 *	D	TRANSPORTER, SLUFAC
KA2350 *	C	DIGITAL TOPOGRAPHIC SPT SY (DTSS)
KA2351 *	C	QUICK RESPONSE MULTI COLOR PRINTER
M15300 *	B	E DET SET, MINE, MET/NON-MET, AN/PRS-B
M17900 *	D	DETECTING SET, MINE, AN/PRS-12
M96300	C	LAUNCHER, SLUFAC
R21000 *	A	VEHICLEMAGNETIC SIGNATURE DUP
R22300 *	B	MINEFIELD RECONNAISSANCE AND DETECTOR SYS (M
WC5112 *	C	COUNTEROBSTACLE VEHICLE (COV)
X00600 *	C	MINE CLEARING ROLLER

***** U N C L A S S I F I E D *****

Tue Jul 29 1986 12:14:05

`fundsysbdp`

This report produces an unclassified listing of system cumulative RDTE funding for development systems in a mission area. Systems are listed in BDP priority order, with base case systems first.

`expr [first pg#] | acego ../neil/fundsysbdp [pg letter] [code] [name]`

***** U N C L A S S I F I E D *****

EMW I-N DEVELOPMENT SYSTEM BDP RANKING
WITH FUNDED RDT&E WORKPACKAGE TOTALS SHOWN

SSN	DA	MA	AMC	MGR	TITLE	FY87	FY88	FY89	FY90	FY91	FY92	RATING
1.	3H2000	CC	TACOM		ARMORED FAMILY OF VEHICLES (AFV)	80710	127413	183608	266537	529400	535190	3560.00
2.	3T0003	CS	BELVOIR		AIRBORNE MINEFIELD DETECTION & RECONN SYS--A	10674	6128	6950	12500	9200	9337	3332.00
3.	3H2018	CS	TACOM		AFV-COMBAT MOBILITY VEHICLE (CMV)	24141	9939	14429	21873	19189	19834	2533.40
4.	KA2950	CSS	BELVOIR		DIGITAL TOPOGRAPHIC SPT SY (DTSS)	0	0	0	0	0	0	2348.40
5.	R21000	CS	PH-MCD		VEHICLE MAGNETIC SIGNATURE DUP	5913	5257	5084	2600	1050	0	2331.60
6.	3H2015	CS	TACOM		AFV-ENGINEER SQUAD (SAPPER)	15943	3487	5300	8577	3500	3400	2221.80
7.	3H2016	CS	TACOM		AFV-COMBAT EXCAVATOR (CEV)	0	0	0	0	0	0	1985.60
8.	3T0078	CSS	PH-MEP		15/30/60 KW SIGNATURE SUPPRESSED GEN SETS	4435	0	0	0	0	0	1903.60
9.	3H2017	CS	TACOM		AFV - COMBAT EARTHROVER (CEM)	0	0	0	0	0	0	1859.20
10.	3H2014	CS	TACOM		AFV-IMPROVED HEAVY ASSAULT BRIDGE (IHAB)	9303	7567	7135	10196	11196	12893	1843.80
11.	R22300	CS	PH-MCD		MINEFIELD RECONNAISSANCE AND DETECTOR SYS (M)	5124	2678	5100	7000	4116	0	1782.00
12.	3H3000	CC	TACOM		FUTURE FAMILY OF VEHICLES (FFOV)	80797	102950	112780	119155	131345	110229	1778.40
13.	3T0037	CS	BELVOIR		WIDE-AREA NEUTRALIZATION MINE DEVICE (WAND)	0	0	450	1548	1900	0	1767.80
14.	3T0020	CSS	BELVOIR		INTEGRATED POWER ENVIRONMENTAL CONTROL SYSTE	2877	5272	3371	2416	3020	1898	1666.80
15.	6E0001	CSS	LABCOM		ADVANCED TACTICAL POWER SOURCES	10986	13617	12628	13019	13244	22470	1649.60
16.	KA2351	CSS	BELVOIR		QUICK RESPONSE MULTI COLOR PRINTER	5127	5441	5829	2633	5560	4288	1448.00
17.	3T0304	CS	PH-MEP		SURVIVABLE TACTICAL ARMY GENERATOR	0	0	0	0	0	0	1288.40
18.	3T0048	CS	BELVOIR		LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (L	0	0	440	2500	0	0	1236.60
19.	6E0002	CSS	LABCOM		THERMOELECTRIC GENERATORS	937	994	0	0	0	0	1159.40
20.	3T0300	CSS	BELVOIR		GEN INTEG SYS FOR TACTICAL VEHICLES	0	0	1034	737	678	2495	1133.20
21.	HC5112	CS	BELVOIR		COUNTEROBSTACLE VEHICLE (COV)	13580	11696	12176	13804	13824	19398	1130.70
22.	3T0049	CS	BELVOIR		IMPROVED LINE CHARGE	2904	2840	2256	2321	4320	2683	1098.20
23.	6T0007	CS	BELVOIR		COUNTERMINE CONCEPTS - BROAD BASED TECHNICAL	19055	30898	71262	71693	73382	77736	1075.20
24.	R63300	CSS	BELVOIR		THERMOELECTRIC GENERATOR 1.5KW	937	994	0	0	3900	2505	1066.60
25.	3T0031	CS	BELVOIR		IMPROVED HAND HELD MINE DETECTOR	5828	5823	9279	6039	6714	7616	1054.60
26.	3H0201	CS	TACOM		SEE VARIANT	6587	6762	7387	9105	9157	14630	999.80
27.	3T0112	CS	BELVOIR		FIELD ARMY MAPPING SYSTEMS (DTSS)	0	0	0	0	0	0	993.20
28.	3T0306	CS	PH-MEP		INTEGRATED VEHICLE/GENERATOR	0	0	0	0	0	1103	934.80
29.	E37301	C6S	PH-MCD		TIME DELAY FIRING DEVICE (TDFD)	710	1276	1977	0	0	0	883.60
30.	3T0042	C6S	BELVOIR		GROUNDWATER DETECTION SYSTEM	0	447	500	0	0	0	826.80
31.	3H2032	CS	TACOM		AFV - MINE DISPERSING VEHICLE	9797	212	14551	13412	8837	12809	806.60
32.	3T0305	CS	PH-MEP		ARMY 21 GENERATOR SET	0	0	0	0	0	1600	764.00
33.	6T0017	C6S	PH-MEP		SYSTEMS & TECHNICAL PERFORMANCE ASSESSMENT	1500	1416	1305	1244	1820	1800	755.40
34.	3T0037	CS	BELVOIR		AUTOMATED COUNTERBARRIER SYSTEM	0	0	0	0	0	0	752.00
35.	3T0013	CSS	PH-MEP		10KW SILENT POWER PLANT	395	1523	3191	2000	0	0	715.40
36.	3T0019	CSS	PH-MEP		0.5 - 10KW LOW NOISE GENERATOR SETS	119	0	0	0	0	0	720.80
37.	3T0079	CSS	BELVOIR		3KW FREE PISTON STIRLING ENGINE GENERATOR SE	650	200	0	766	1916	0	696.80
38.	E37201	CS	PH-MCD		LINEAR SHAPED CHARGE	350	220	0	0	0	0	685.00
39.	3T0301	CSS	BELVOIR		ADVANCED GENERATOR SET COMPONENTS	0	0	0	0	0	800	625.40
40.	3T0028	CS	BELVOIR		CANETIP	0	0	2091	6731	5612	6234	606.80
41.	5T0007	CS	BELVOIR		MULTIPURPOSE DETECTION SYSTEM	0	0	637	4290	3236	6200	606.80
42.	E37401	CSS	PH-MCD		PENETRATION AUGMENTATION MUNITION (PAM)	1872	2370	2760	3100	4360	1000	592.60
43.	6D0016	CS	PH-MCD		SELECTABLE LIGHTWEIGHT ATTACK MUNITION (SLAM)	0	0	0	0	0	0	589.80
44.	HC102C	CS	BELVOIR		HEAVY ASSLT BRIDGE (HAB)	18053	16239	22459	29761	32740	35987	587.20
45.	6D0002	CS	PH-MCD		WALL DEMOLITION MUNITION	3250	3500	3500	4046	4854	5232	565.20

***** U N C L A S S I F I E D *****
Tue Jul 29 1986 11:21:37

fundwpbdp

This report produces an unclassified listing of workpackages for a mission area with their funded amounts. Workpackages are listed in BDP priority order. Sorted by fund subcategory. (6.3, 6.4, and 6.7 only).

expr [first pg#] | acego ../neil/fundwpbdp [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

EMH 1 TO N 6.3A WORKPACKAGE BDP RANKINGS
WITH FUNDED FISCAL RESOURCES SHOWN

PE/PROJ/TASK/WMPKO	TITLE	FUNDED DEVELOPMENT SCHEDULE												
		FY85	FY86	FY87	FY88	FY89	FY90	FY91	FY92					
1.	WIDE AREA NEUTRALIZATION DEVICE (HAND)													
	BELVOIR 63606 D608 16 1023													
2.	COUNTEROBSTACLE VEHICLE													
	BELVOIR 63606 D608 08 1006													
3.	BARRIER 21/ANTI-TANK OBSTACLE													
	BELVOIR 63606 D608 01 1054													
4.	AIRBORNE MINEFIELD DETECTION & RECONNAISSANCE SYSTEM													
	BELVOIR 63606 D608 11 1007													
5.	MINEFIELD IMAGE RECOGNITION & CUEING STATIONS (MIRACS)													
	BELVOIR 63606 D608 48 1315													
6.	CANE TIP													
	BELVOIR 63606 D608 05 1005													
7.	UNFINANCED COUNTERMINE PROGRAM													
	BELVOIR 63606 D608 27 1340													
8.	UNFINANCED COUNTERMINE PROGRAM - INCREMENT													
	BELVOIR 63606 D608 27 1341													
9.	SCATTERABLE MINEFIELD DETECTION SYSTEM (SCM)													
	BELVOIR 63606 D608 30 1012													
10.	COMBAT ENGINEER COMPONENTS													
	BELVOIR 63102 D601 01 1047													
11.	MAN-MADE OBSTACLE REPORTING AND DISPOSITION SYSTEM (MOR)													
	BELVOIR 63606 D608 02 1079													
12.	IMPROVED LINE CHARGE													
	BELVOIR 63606 D608 25 1009													
13.	AUTOMATED COUNTER BARRIER SYSTEM													
	BELVOIR 63606 D608 32 1013													
14.	WIDE AREA NEUTRALIZATION DEVICE (HAND) - INCREMENT													
	BELVOIR 63606 D608 16 1305													
15.	TACTICAL OBSTACLE MARKING SYSTEM (TOMS)													
	BELVOIR 63606 D608 31 1067													
16.	REAR AREA MINE CLEARING SYSTEM (RAHV)													
	BELVOIR 63606 D608 15 1074													
17.	ADVANCED MECHANICAL MINE CLEARING DEVICE													
	BELVOIR 63606 D608 33 1014													
18.	MINE COMPONENTS (MINIMUM BAND)													
	ARDEC 63606 D006 02 03347													
19.	MINE COMPONENTS (AT RISK BAND)													
	ARDEC 63606 D006 02 13347													
20.	LANDMINE WARFARE													
	ARDEC 63606 D006 02 93347													

***** U N C L A S S I F I E D *****

Tue Jul 29 1986 11:25:29

fundwpbdp2

This report produces an unclassified listing of workpackages for a mission area with their funded amounts. Workpackages are listed in BDP priority order. (6.3, 6.4, and 6.7) Extra spacing in the report is provided.

expr [first pg#] i acego ../neil/fundwpbdp2 [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

EMW 1 TO N WORKPACKAGE BDP RANKINGS
WITH FUNDED FISCAL RESOURCES SHOWN

PE/PROJ/TASK/WKPKG	TITLE	FUNDED DEVELOPMENT SCHEDULE										
		FY85	FY86	FY87	FY88	FY89	FY90	FY91	FY92			
1.	MINEFIELD RECONNAISSANCE AND DETECTOR SYSTEM (MIRADOR) ARDEC 63619 D606 27 01343		4047	5124	2678							
2.	MINEFIELD RECONNAISSANCE AND DETECTOR SYSTEM (MIRADOR) ARDEC 64612 D415 10 01340				5100	7000	3500					
3.	15/30/60 KW SIGNATURE SUPPRESSED GENERATOR SETS MEP 64714 D194 15 7008	2200	5146	6656	4435							
4.	MULTI-PURPOSE DETECTION SYSTEM AD (MIN) ARDEC 63619 D606 37 01351					1577						3400
5.	HEAVY ASSAULT BRIDGE BELVOIR 63726 D001 01 1025	3007	6658	3400	1637							
6.	MINEFIELD RECONNAISSANCE AND DETECTOR SYSTEM (MIRADOR) ARDEC 63619 D606 27 91343											
7.	MULTI-PURPOSE DETECTION SYSTEM ARDEC 63619 D606 37 91351											
8.	WIDE AREA NEUTRALIZATION DEVICE (HAND) BELVOIR 63606 D608 16 1023				450	1548	1900					
9.	WIDE AREA NEUTRALIZATION DEVICE (HAND) (MIN) ARDEC 64612 D415 12 01355											
10.	COUNTEROBSTACLE VEHICLE BELVOIR 63606 D608 CB 1006	1933	2230	1400	100							
11.	BARRIER 21/ANTI-TANK OBSTACLE BELVOIR 63606 D608 01 1054											
12.	COUNTEROBSTACLE VEHICLE (COV) ARDEC 63619 DLO1 31 91346											
13.	TACTICAL EXPLOSIVE SYSTEM (TEXS) ED (MIN) ARDEC 64612 D021 02 01325			6700	250							
14.	AIRBORNE MINEFIELD DETECTION & RECONNAISSANCE SYSTEM BELVOIR 63606 D608 11 1007		3417	3850	1350	1250						

***** U N C L A S S I F I E D *****

Tue Jul 29 1986 11:26:01

fundwplrp

This report produces an unclassified listing of workpackages for a mission area with their funded amounts. Workpackages are listed in PDIP priority order. (6.3, 6.4, and 6.7) Extra spacing in the report is provided.

expr [first pg#] ; acego ../neil/fundwplrp [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

EMW 1 TO N WORKPACKAGE PDIP RANKINGS
WITH FUNDED FISCAL RESOURCES SHOWN

PE/PROJ/TASK/WPKG TITLE		FUNDED DEVELOPMENT SCHEDULE									
		FY85	FY86	FY87	FY88	FY89	FY90	FY91	FY92		
1.	COMBAT ENGINEER COMPONENTS BELVOIR 63102 DJO1 01 1047		776			938	4113	4651	1717		
2.	HEAVY DRY SUPPORT BRIDGE BELVOIR 63102 DJO1 02 1317			1559	3428	2500					
3.	HEAVY DRY SUPPORT BRIDGE - INCREMENT BELVOIR 63102 DJO1 02 1311										
4.	MINE COMPONENTS (MINIMUM BAND) ARDEC 63606 D006 02 03347			1675	212	3332	3556	1818	2817		
5.	BARRIER 21/ANTI-TANK OBSTACLE BELVOIR 63606 D608 01 1054										
6.	MAN-MADE OBSTACLE REPORTING AND DISPOSITION SYSTEM (MOR) BELVOIR 63606 D608 02 1079		300								
7.	VEHICLE HARDENING BELVOIR 63606 D608 04 1066										
8.	VEHICLE HARDENING - INCREMENT BELVOIR 63606 D608 04 1301										
9.	CANE TIP BELVOIR 63606 D608 05 1005		1250	1547		2091	3223	2000			
10.	GENERIC TECHNOLOGY BASE - COUNTERMINE FLAILS BELVOIR 63606 D608 06 1124		580								
11.	LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (LIMBS) BELVOIR 63606 D608 07 1309					440	2500	750			
12.	LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (LIMBS)-IMCRE BELVOIR 63606 D608 07 1310										
13.	COUNTEROBSTACLE VEHICLE BELVOIR 63606 D608 08 1006		1933	2250	1400	100					
14.	AIRBORNE MINEFIELD DETECTION & RECONNAISSANCE SYSTEM BELVOIR 63606 D608 11 1007			3417	3850	1350	1250				

***** U N C L A S S I F I E D *****

Tue Jul 29 1986 11:26:43

lrrdp1

This report produces an unclassified diagnostic report listing all the ssns and system titles that exist in the lrrdp file in alphabetical order. It can be used to identify duplicate system entries between ssns and nsis.

acego ../neil/lrrdp1

LRRDP SYSTEMS BY TITLE

1L0004	FS	WELDING MACHINE ELEC ARC 350 AMP (CC-C
3A010B	AVN	0.5 - 10KW LOW NOISE GENERATOR SETS
3M5006	CSS	0.3KW THERMOELECTRIC GENERATOR
G22B00	NBC	1.5/3KW POWER CONDITIONING MODULE
NA012B		100 TON CRANE
M63300	CSS	103MM TANK GUN ENHANCEMENT
3T0019	EMW	10KW SILENT POWER PLANT
R61500	EMW	120MM XM929 SMK CTG FOR THE BATTALION MORTAR SYSTEM (BMS)
3T0030	EMW	15/30/60 KW SIGNATURE SUPPRESSED GEN SETS
R32400	CSS	230-GALLON EXTERNAL FUEL TANK
6A0675		2500-SHAFT HORSEPOWER (SHP) ENGINE
3D5134	CCL	30 & 60KW POWER CONDITIONERS
3T007B	EMW	3KW FREE PISTON STIRLING ENGINE GENERATOR SET
3A0094	AVN	40 MM HIGH VELOCITY SMOKE GRENADE
3A0073	AVN	40.000LB INTRADEPOT AMMUNITION TRANSPORTER
3T0023	EMW	42.000 LB AIRDROP SYSTEM
3T0079	EMW	4TH PSYOP GROUP
3D5937	CCL	60MM, XM722 SMK CTG FOR THE LTHT COMPANY MORTAR SYS (LWCMS)
3T0026	CSS	65 TON CRANE
4T5022	CSS	9HW
8B2050	SOF	A/C DECONTAMINATION, DEICING, AND CLEANING SYSTEM (ADDCS)
3D5931	CCL	A/C NBC DETECTION SYSTEMS
X00900	CSS	AAM CB MASK
G23405		ABN ADDS
3A0037	AVN	ABN NAVSTAR
3A0021	AVN	ABN PLRS
3A0017	CC	ABN SINCOARS (1B6)
AZ3503		ACOUSTIC DETECTION SYSTEM (ADS)
K48000		ACA AVIONICS INTEGRATION
K20500		ACATT
AZ3500	COM	ACCELERATED FUEL QUALIFICATION PROCEDURE
3C1007	SOF	ACCB SYSTEMS ENGINEERING
6A0049	AVN	ACFT 9HW
3B2A01	AVN	ACTUATOR, AT MINE M1
3T0030	CSS	AD/CC2 MODS
3C0003	CC	ADA
AZ3405	AVN	ADPE FOR NON TAC MGT INFO SYS
E37600	AD	ADV ANTITANK WEAPONS SYS MED (AAMS-M)
CE9000		ADV FIELD ARTILLERY TACT DATA SYS
8B2A04	AVN	ADV HIGH SPEED RETROCKET SYSTEM
BE4150		ADV MATERIAL EXTERNAL CARGO SLING SYSTEM (AMCSS)
C99300	CCL	ADV MISSION PLANNING STATION (AMPS)
B2B600	FS	ADV POWER SOURCE FOR FUEL CELL
4T5026	CSS	ADV THREAT ALQ-136 (SEMA)
3A0008	CSS	ADV THREAT ALQ-144A IR JAMMER
6A0046	AVN	ADV THREAT AN/ALQ-136A(V)1 RADAR JAMMER
3T0075	EMW	ADV THREAT AT/ALG-162 CW RADAR JAMMER
3A0031	AVN	ADV. LASER MARKERS
3A0041	AVN	ADVANCED AIRBORNE RADIAC
3A0033	AVN	ADVANCED ANTITANK GSE SYSTEM
3A0035	AVN	ADVANCED AT WEAPON
3C0004	SOF	ADVANCED AT WEAPON
WC5216	CS	
C99400		
WE2230	CCL	
WE223A	CCH	

lrrdppri

This report produces a confidential listing of all the pdip/increments and their priorities that are currently contained in the lrrdppri file.

expr [first pg#] ; acego ../neil/lrrdppri [pg letter]

***** S A M P L E *****

DA PRIORITY BY PDIP AND INCREMENT

PDIP/INCREMENT	PRIORITY
186N-01	0.00
256C-01	0.00
298H-01	0.00
288J-02	0.00
288K-02	0.00
288M-01	0.00
288Q-01	0.00
288Q-02	0.00
288S-01	0.00
28A3-01	0.00
28A3-02	0.00
309E-01	0.00
307M-01	0.00
3R70-01	0.00
3R70-02	0.00
3R70-03	0.00
3R7H-01	0.00
3R7J-01	0.00
384U-01	0.00
384F-01	0.00
387E-01	0.00
387E-02	0.00
387E-04	0.00
38CD-01	0.00
38X6-01	0.00
38X6-02	0.00
38X6-03	0.00
3M8M-01	0.00
483F-01	0.00
486A-01	0.00
4868-01	0.00
486V-01	0.00
486Y-01	0.00
48Z3-01	0.00
48Z3-02	0.00
578D-01	0.00
5M70-01	0.00
5M7Y-01	0.00
5P7M-01	0.00
5R5C-01	0.00
584T-01	0.00
586A-01	0.00
586X-01	0.00
587H-01	0.00
587W-01	0.00
588M-01	0.00
588M-02	0.00

***** S A M P L E *****

G- 1

Thu Aug 7 1986 11:51:37

lrrdpsys

This report produces an unclassified or confidential listing of all development systems in a mission area, in PDIP increment priority. If a system appears in more than one increment, only the highest priority increment will be listed. All base case systems appear at the top of the list.

expr [first pg#] i acego ../neil/lrrdpsys [pg letter] [code] [name] [class]

***** U N C L A S S I F I E D *****

DA DEVELOPMENT SYSTEM PRIORITY FOR EMW

	SSN	DA MA	AMC MGR	TITLE	PDIP	PRIORITY
1.	R04800	CS	TACOM	SMALL EMPACEMENT EXCAVATOR (SEE)	48AA-01	base case
2.	1M1006	CC	TACOM	COMBAT ENGINEER VEHICLE	FL61-01	base case
3.	8B1100	CSS	LABCOM	BATTERY CHARGER PP-7286/U	6V3H-01	base case
4.	R03000	CS	TACOM	TRUCK, DUMP, 20T (CCE)	6S5E-01	base case
5.	M05100	CS	TACOM	TRACTOR, FULLTRCAD LOW SPEED DSL	6MBF-02	base case
6.	M06000	CS	TACOM	TRACTOR, FL. TRKD, LS, D SP, SECT, AT	6MBF-02	base case
7.	R14201	CS	TACOM	SCRAPER ELEVATING SP 9 CU YD MIN SEC	6MBF-02	base case
8.	6D0016	CS	PM-MCD	SELECTABLE LIGHTWEIGHT ATTACK MUNITION (SLAM)	6RBP-01	base case
9.	3T0097	CSS	BELVOIR	INTEGRATED CHEMICAL FILTER ENVIRONMENTAL EQUIP-ICE	6M70-01	base case
10.	G23000	CC	TACOM	ARMORED VEH LAUNCH BRIDGE (AVLB) - PIP	FL61-01	base case
11.	5T0027	CS	BELVOIR	COMBAT ENGINEER COMPONENTS	6RBB-01	base case
12.	5T0036	CS	BELVOIR	HIGH MOBILITY BRIDGING	6RBB-01	base case
13.	3T0046	CS	BELVOIR	ADVANCED MINE CLEARING SYS	6M7L-01	base case
14.	3T0048	CS	BELVOIR	LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (LIMBS)	6M7L-01	base case
15.	3T0049	CS	BELVOIR	IMPROVED LINE CHARGE	6M7L-01	base case
16.	3T0051	CS	BELVOIR	TACTICAL OBSTACLE MARKING SYSTEM (TOMS)	6M7L-01	base case
17.	3T0055	CS	BELVOIR	REACTIVE FRAGMENT MINE NEUTRALIZATION SYSTEM (FRAG)	6M7L-01	base case
18.	3T0056	CS	BELVOIR	WIDE-AREA NEUTRALIZATION MINE DEVICE (WAND)	6M7L-01	base case
19.	3T0057	CS	BELVOIR	MAGNETIC SIGNATURE SILENCING SYSTEM (MASSIS)	6M7L-01	base case
20.	3T0058	CS	BELVOIR	ELECTRIC EXPLOSIVE DETONATION SYSTEM	6M7L-01	base case
21.	3T0307	CS	BELVOIR	AIRBORNE MINEFIELD DETECTION & RECONN SYS--AMIDARS	6M7L-01	base case
22.	3T0003	CS	BELVOIR	SCATTERMINE RADAR (SCMR)	6M7L-01	base case
23.	3T0006	CS	BELVOIR	MULTIPURPOSE DETECTION SYSTEM	6M7L-01	base case
24.	5T0007	CS	BELVOIR	CANETIP	6M7L-01	base case
25.	5T0028	CS	BELVOIR	AUTOMATED COUNTERBARRIER SYSTEM	6M7L-01	base case
26.	5T0037	CS	BELVOIR	VEHICLE HARDENING	6M7L-01	base case
27.	5T0038	CS	BELVOIR	REAR AREA MINE CLEARING SYSTEM (RAMV)	6M7L-01	base case
28.	5T0040	CS	BELVOIR	MAN MADE OBSTAC RPT & DUP SYS (MORDS)	6M7L-01	base case
29.	5T0041	CS	BELVOIR	COUNTERMINE CONCEPTS - BROAD BASED TECHNICAL AREA	6M7L-01	base case
30.	6T0007	CS	BELVOIR	BARRIERS-BROAD BASE TECH AREA	6M7L-01	base case
31.	6T0025	CS	BELVOIR	PENETRATION AUGMENTATION MUNITION (PAM)	6M7L-01	base case
32.	E57401	CSS	PM-MCD	BRIDGE & ROAD MUNITION (BRM)	6M7L-01	base case
33.	E57501	CSS	PM-MCD	EXPLOSIVE FOX HOLE DIQOER (EXFOD)	6M7L-01	base case
34.	E77401	CS	PM-MCD	M607 PIP	6M7L-01	base case
35.	EA3800	CS	PM-MCD	GEMSS AUX MINE DISPNSR XM138 (FLIPPER)	6M7L-01	base case
36.	Q27000	CS	AMCCOM	DETECTING SET, MINE, AN/PSS-12	6M7L-01	base case
37.	M17900	CS	BELVOIR	VEHICLEMAGNETIC SIGNATURE DUP	6M7L-01	base case
38.	R21000	CS	PM-MCD	MINEFIELD RECONNAISSANCE AND DETECTOR SYS (M)	6M7L-01	base case
39.	R22300	CS	PM-MCD	IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)	6M7L-01	base case
40.	W81038	CS	PM-MCD	COUNTEROBSTACLE VEHICLE (COV)	6M7L-01	base case
41.	4C5112	CS	BELVOIR	RIBBON BRIDGE, IMPROVED	6XES-01	base case
42.	1T0012	CS	BELVOIR	TUNNEL DETECTION SYSTEM (TUDS)	6XES-01	base case
43.	3T0002	CS	BELVOIR	HEAVY DRY SUPPORT BRIDGE	6XES-01	base case
44.	3T0006	CS	BELVOIR	OVERHEAD COVER (OHC) FOR INDIV FIGHTING POSITION	6XES-01	base case
45.	3T0060	CC	BELVOIR	DUAL ROLE BRIDGE	6XES-01	base case
46.	3T0066	CS	BELVOIR	TACTICAL BRIDGE ACCESS/EGRESS	6XES-01	base case
47.	M28500	CS	BELVOIR			

***** U N C L A S S I F I E D *****

Fri Jul 11 1986 16:17:17

lrrdpwp

This report produces an unclassified or confidential listing of all workpackages in a mission area, in PDIP increment priority.

expr [first pg#] : acego ../neil/lrrdpwp [pg letter] [code] [name] [class]

***** U N C L A S S I F I E D *****

DA WORKPACKAGE PRIORITY FOR EMW

PE	PROJ TASK	NO	TITLE	COMMAND	PDIP	PRIORITY
1.	63102 D001 01	1047	COMBAT ENGINEER COMPONENTS	BELVOIR	6R8B-01	
2.	63102 D001 02	1317	HEAVY DRY SUPPORT BRIDGE	BELVOIR	6R8B-01	
3.	63102 D001 02	1318	HEAVY DRY SUPPORT BRIDGE - INCREMENT	BELVOIR	6R8B-01	
4.	63606 D006 02	03347	MINE COMPONENTS (MINIMUM BAND)	ARDEC	6M7L-01	
5.	63606 D608 01	1054	BARRIER 21/ANTI-TANK OBSTACLE	BELVOIR	6M7L-01	
6.	63606 D608 02	1079	MAN-MADE OBSTACLE REPORTING AND DISPOSITION SYSTEM (MORDS)	BELVOIR	6M7L-01	
7.	63606 D608 04	1066	VEHICLE HARDENING	BELVOIR	6M7L-01	
8.	63606 D608 04	1301	VEHICLE HARDENING - INCREMENT	BELVOIR	6M7L-01	
9.	63606 D608 05	1005	CANE TIP	BELVOIR	6M7L-01	
10.	63606 D608 06	1124	GENERIC TECHNOLOGY BASE - COUNTERMINE FLAILS	BELVOIR	6M7L-01	
11.	63606 D608 07	1309	LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (LIMBS)	BELVOIR	6M7L-01	
12.	63606 D608 07	1310	LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (LIMBS)-INCREMENT	BELVOIR	6M7L-01	
13.	63606 D608 08	1006	COUNTEROBSTACLE VEHICLE	BELVOIR	6M7L-01	
14.	63606 D608 11	1007	AIRBORNE MINEFIELD DETECTION & RECONNAISSANCE SYSTEM	BELVOIR	6M7L-01	
15.	63606 D608 12	1055	SARAMID	BELVOIR	6M7L-01	
16.	63606 D608 13	1304	COUNTERMINE TECH BASE	BELVOIR	6M7L-01	
17.	63606 D608 14	1057	REACTIVE FRAGMENT MINE NEUTRALIZATION SYSTEM	BELVOIR	6M7L-01	
18.	63606 D608 14	1303	REACTIVE FRAGMENT - INCREMENT	BELVOIR	6M7L-01	
19.	63606 D608 15	1074	REAR AREA MINE CLEARING SYSTEM (RAMV)	BELVOIR	6M7L-01	
20.	63606 D608 16	1023	WIDE AREA NEUTRALIZATION DEVICE (WAND)	BELVOIR	6M7L-01	
21.	63606 D608 16	1305	WIDE AREA NEUTRALIZATION DEVICE (WAND) - INCREMENT	BELVOIR	6M7L-01	
22.	63606 D608 25	1009	IMPROVED LINE CHARGE	BELVOIR	6M7L-01	
23.	63606 D608 27	1340	UNFINANCED COUNTERMINE PROGRAM	BELVOIR	6M7L-01	
24.	63606 D608 27	1341	UNFINANCED COUNTERMINE PROGRAM - INCREMENT	BELVOIR	6M7L-01	
25.	63606 D608 29	1049	IMPROVED HAND HELD MINE DETECTOR	BELVOIR	6M7L-01	
26.	63606 D608 29	1300	IMPROVED HAND HELD MINE DETECTOR - INCREMENT	BELVOIR	6M7L-01	
27.	63606 D608 30	1012	SCATTERABLE MINEFIELD DETECTION SYSTEM (SCM)	BELVOIR	6M7L-01	
28.	63606 D608 31	1067	TACTICAL OBSTACLE MARKING SYSTEM (TOMS)	BELVOIR	6M7L-01	
29.	63606 D608 31	1306	TOMS - INCREMENT	BELVOIR	6M7L-01	
30.	63756 D608 32	1013	AUTOMATED COUNTER BARRIER SYSTEM	BELVOIR	6M7L-01	
31.	63606 D608 33	1014	ADVANCED MECHANICAL MINE CLEARING DEVICE	BELVOIR	6M7L-01	
32.	63606 D608 33	1307	ADVANCED MECHANICAL MINE CLEARING DEVICE - INCREMENT	BELVOIR	6M7L-01	
33.	63606 D608 36	1081	MULTIPURPOSE DETECTION SYSTEM	BELVOIR	6M7L-01	
34.	63606 D608 48	1315	MINEFIELD IMAGE RECOGNITION & CUEING STATIONS (MIRACS)	BELVOIR	6M7L-01	
35.	63606 D608 49	1302	ADVANCED TECHNOLOGY PORTABLE MINE DETECTOR - INCREMENT	BELVOIR	6M7L-01	
36.	63606 D608 49	1314	ADVANCED TECHNOLOGY PORTABLE MINE DETECTOR - INCREMENT	BELVOIR	6M7L-01	
37.	63619 D003 02	01312	ICOMS AD EFFORT (MIN)	ARDEC	6M7L-01	
38.	63619 D003 04	01303	VOLCANO AD EFFORT (MIN)	ARDEC	6M7L-01	
39.	63619 D003 05	01315	WIDE AREA MINE (WAM) AD EFFORT (MIN)	ARDEC	6M7L-01	
40.	63619 D003 05	01374	ADVANCED MINE OBSTACLE SYSTEM AD (MIN)	ARDEC	6M7L-01	
41.	63619 D003 05	01372	SELECTABLE LIGHTWEIGHT ATTACK MUNITION (SLAM) AD (MIN)	ARDEC	6M7L-01	
42.	63619 D003 05	01319	CONTROLABLE MINE SYSTEM AD EFFORT (MIN)	ARDEC	6M7L-01	
43.	63619 D606 27	01343	MINEFIELD RECONNAISSANCE AND DETECTOR SYSTEM (MIRADOR) (MIN)ARDEC	ARDEC	6M7L-01	
44.	63619 D606 37	01351	MULTI-PURPOSE DETECTION SYSTEM AD (MIN)	ARDEC	6M7L-01	
45.	64612 D413 03	01339	VEHICLE MAGNETIC SIGNATURE DUPLICATOR (VEMASID) (MIN)	ARDEC	6M7L-01	
46.	64612 D413 04	01360	VEHICLE MAGNETIC SIGNATURE DUPLICATOR (VEMASID) P31 (MIN)	ARDEC	6M7L-01	
47.	64612 D413 05	01353	UNIVERSAL PORTABLE MINE DETECTOR (UPMD) AD (MIN)	ARDEC	6M7L-01	

***** U N C L A S S I F I E D *****

mamppdip

This report produces a confidential listing of the pdips, priorities, systems, and workpackages appropriate to a single mission area. It is our first attempt to recreate the Irrdap out of the mamp database.

expr [first pg#] : acego ../neil/mamppdip [pg letter] [code] [name]

***** S A M P L E *****

MAMP PDIP SUMMARY FOR EMW

PDIP: 6XM6 Title: TOPO SPT EQUIPMENT DA/TRADOC Mission Areas: CS / EMW Proponent: ENS

SYSTEM AND RELATED PROJECT TITLES

PROCUREMENT/RDTE SCHEDULE

	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	BDP
Increment: 6XM6-01 Priority: 0.00																	
*** Development Systems ***																	
K42350 DIGITAL TOPOGRAPHIC SPT SY (DTSS)																	2348
K42351 QUICK RESPONSE MULTI COLOR PRINTER																	1448
63712 D580 BELVOIR FIELD ARMY MAPPING																	
64716 D579 QUICK REACTION MULTICOLOR PRINTER																	
3T0112 FIELD ARMY MAPPING SYSTEMS (DTSS)																	993
Increment: 6XM6-02 Priority: 0.00																	
*** Development Systems ***																	
K42351 QUICK RESPONSE MULTI COLOR PRINTER																	1448
Increment: 6XM6-03 Priority: 0.00																	
*** Development Systems ***																	
K42350 DIGITAL TOPOGRAPHIC SPT SY (DTSS)																	2348

Legend: ***/^^^ - Funded/Unfunded Procurement
+ + / - - - - - Funded/Unfunded RDTE

***** S A M P L E *****

Thu Aug 7 1986 11:29:33

pdipcom

This unclassified report summarizes the systems in a mission area by the commodity line that they belong to. It is like the commodity summary, but it is intended to be used to help structure future pdip submissions

expr [first pg#] : acego ../neil/pdipcom [pg letter] [code] [name] [class]

***** U N C L A S S I F I E D *****

AIR DROP PDIP COMMODITY SUMMARY FOR CSS

Acquisition Strategy THIS LINE CONTAINS SYSTEMS FOR AIRDROP OF PERSONNEL (MASS ASSAULT AND SOF), WEAPONS SYSTEMS AND HEAVY EQUIPMENT/VEHICLES AND SUPPLIES (FOOD, AMMO, FUEL) DEEP BEHIND ENEMY LINES WHERE CONVENTIONAL MEANS OF DELIVERY/RE-SUPPLY ARE IMPOSSIBLE TO EMPLOY. DEVELOPMENTAL SYSTEMS WILL ALLOW AIRDROP TO BE CONDUCTED AT MUCH LOWER ALTITUDES (1200 TO 300 FT) THEREBY REDUCING AIRCRAFT VULNERABILITY, INCREASED PAYLOAD UP TO 60,000 LBS (AS COMPARED TO 35,000 LBS AT PRESENT), REDUCED RIGGING TIME AND LABOR AND ELIMINATOR OF PAPER HOMEYCOMB THRU SOFT LANDING SYSTEMS AND DIVE-DN/DIVE-OFF CAPABILITY, REDUCED

SYSTEM TITLES AND BDP DEFICIENCIES

		PROCUREMENT/RDTE SCHEDULE																	
		PDIP	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00	01	PRI/BDP
4TS006	**** DEVELOPMENT ***** AIRDROP CONTROLLED EXIT SYS (LINKED PLAT) NATICK 19-B 62-D 70-C 75-B 97-D 131-C 255-A CSS / CSS	6RBP-02																	1721
4TS022	42,000 LB AIRDROP SYSTEM NATICK 62-C 75-B 82-C 97-D 131-C 332-C CSS / CSS	6RBP-02																	978
4TS019	AIRDROP SYSTEMS FOR C17 AIRCRAFT NATICK 62-C 75-B 145 236-B 279-B CSS / CSS	6RBP-02																	903
4TS013	HEAVY DROP RIGGING SYSTEM FOR CBT VEHICLES NATICK 62-C 75-B 131-C 236-C 279-C CSS / CSS	6RBP-02																	857
4TS003	LOW ALT TACT ASSAULT PERSONNEL PARACHUTE SYS NATICK 62-C 75-A CSS / CSS	6RBP-01																	855
4TS026	ADV HIGH SPEED REIROCKET SYSTEM NATICK 75-B 77-C 93-C 145 332-C CSS / CSS	6RBP-01																	739
4TS014	LOW ALTITUDE RETROCKET AIRDROP SYSTEM NATICK 75-B 77-C 93-C 278 313 CSS / CSS	6RBP-02																	739
4TS012	AIR DELIV SYS FOR HIGH CONSUMP CBT SUPPLIES NATICK 19-C 70-C 73-D 97-D 131-C CSS / CSS	6RBP-02																	676
4TS009	DROP ZONE ASSEMBLY SYSTEMS NATICK 355-A CSS / CSS	6RBP-01																	232
5TS025	**** TECHNOLOGY DEMONSTRATORS ***** CARGO OFFSET DELIVERY PLATFORM NATICK 62-C 73-C 131-C CSS / CSS	6RBP-02																	494
6TS002	**** BROAD BASE TECHNICAL AREAS ***** AIRDROP TECHNOLOGY NRDEC 19-C 62-D 73-C 75-C 76-C 97-D CSS / CSS	6RBP-02																	1399
6TS028	STEALTH AERIAL FIRE SUPPORT WEAPON SYSTEM NATICK CSS / CSS																		0

Legend:
 *** - Funded Procurement
 +++ - Funded RDTE
 --- - Unfunded RDTE

***** U N C L A S S I F I E D *****

Thu Jul 17 1986 16:23:24

pdipcomda

This version of pdipcom includes the workpackages uniquely linked to the systems within a single pdip, and shows funding dollars for the planning years.

```
expr [first pg#] | acego ../neil/pdipcomda [pg letter] [code] [name] [class]
```


***** U N C L A S S I F I E D *****

AIR DROP PDIP COMMODITY DECISION AID FOR CSS
WITH CUMULATIVE TOTALS (RTDE/PROC)

Acquisition Strategy. THIS LINE CONTAINS SYSTEMS FOR AIRDROP OF PERSONNEL (MASS ASSAULT AND SOF), WEAPONS SYSTEMS AND HEAVY EQUIPMENT/VEHICLES AND SUPPLIES (FOOD, AMMO, FUEL) DEEP BEHIND ENEMY LINES WHERE CONVENTIONAL MEANS OF DELIVERY/RE-SUPPLY ARE IMPOSSIBLE TO EMPLOY. DEVELOPMENTAL SYSTEMS WILL ALLOW AIRDROP TO BE CONDUCTED AT MUCH LOWER ALTITUDES (1200 TO 300 FT) THEREBY REDUCING AIRCRAFT VULNERABILITY, INCREASED PAYLOAD UP TO 60,000 LBS (AS COMPARED TO 35,000 LBS AT PRESENT), REDUCED RIGGING TIME AND LABOR AND ELIMINATOR OF PAPER HONEYCOMB THRU SOFT LANDING SYSTEMS AND DIVE-DN/DIVE-OFF CAPABILITY, REDUCED

SSN	TITLE	DA/TDC	MA	PDIP	87	88	89	90	91	92	93
415006	AIRDROP CONTROLLED EXIT SYS (LINKED PLAT		CSS / CSS								
1	62-D 70-C 75-B 97-D 131-C 255-A										
1	63218 D242 04 D242-04 NATICK CRIT: 1				425						
2	ADVANCED PARACHUTE RELEASES FOR CARGO AIRDROP SYST										
1	64218 D279 22 D279-22 NATICK CRIT: 1				676	400					
3	60,000 LB CAPACITY LINKED PLATFORM AIRDROP SYSTEM										
1	64218 D279 28 D279-28 NATICK CRIT: 1				400	20					
4	AIRDROP CONTROLLED EXIT SYSTEM (LINKED PLATFORMS)										
1	64218 D279 27 D279-27 NATICK CRIT: 1				400	500	500	514			
	ADVANCED PARACHUTE RELEASES FOR CARGO AIRDROP SYST										
	1				0	0	0	0	0	0	0
	Cumulative Totals Procurement Funded:				1521	820	500	500	514	0	0
	RTDE Funded:										
615002	AIRDROP TECHNOLOGY		CSS / CSS								
1	19-C 62-D 73-C 75-C 76-C 77-C 97-D										
	Cumulative Totals Procurement Funded:				0	0	0	0	0	0	0
	RTDE Funded:				1521	820	500	500	514	0	0
415022	42,000 LB AIRDROP SYSTEM		CSS / CSS								
1	62-C 75-B 82-C 97-D 131-C 332-C										
1	64218 D279 29 D279-29 NATICK CRIT: 1				50						
	42,000 LB CAPACITY AIRDROP SYSTEM										
	Cumulative Totals Procurement Funded:				0	0	0	0	0	0	0
	RTDE Funded:				1571	820	500	500	514	0	0
415019	AIRDROP SYSTEMS FOR C17 AIRCRAFT		CSS / CSS								
1	62-C 75-B 145- 236-B 279-B										
1	63218 D242 05 D242-05 NATICK CRIT: 1				150	400	500	750	870	1004	1621
2	AIRDROP COMPONENT TECHNOLOGY FOR ADVANCED CARGO SY										
1	63218 D242 2X D242-2X NATICK CRIT: 1					600	926				
3	HIGH SPEED LOW ALTITUDE CDS FOR C17										
1	63218 D242 2X D242-2X NATICK CRIT: 1										
	HIGH SPEED LOW ALTITUDE CDS FOR C17										
	(B34)										

***** U N C L A S S I F I E D *****

Thu Jul 17 1986 16:34:25

Legend:
~ - LRRDAP Priority
+ - BDP Rating Score

pdipsys

This report produces an unclassified crossreference of systems to PDIP increments, for a mission area. Systems are grouped in PDIP order.

exp [first pg#] : acego ../neil/pdipsys [pg letter] [code] [name]

***** S A M P L E *****

PDIP TO EMW SYSTEM CROSS REFERENCE

PDIP	INCR	PRIORITY	SSN	DA MA	TDC MA/PROP	PDIP/SYSTEM TITLE	COMMAND
348W	01	0.00	MAB050	IM	SOF JFK	SOF FORWARD DEPLOYED	BELVOIR
456A	01	0.00	R04800	CS	EMW ENS	ITEMS LESS THAN \$2.0M (CSS-EQ)	BELVOIR
6M70	01	0.00	3T0097	CS	CSS LOG	LUPS	TACOM
6M72	01	0.00	3T0062	CS	EMW ENS	SMALL EMPACEMENT EXCAVATOR (SEE)	TACOM
			3T0063	CS	CSS QMS	CBT SVC SPT EQ RDTE	BELVOIR
			3T0064	CS	EMW ENS	INTEGRATED CHEMICAL FILTER ENVIRONMENTAL EQUIP-ICE	BELVOIR
			3T0065	CS	EMW ENS	RDTE MANAGEMENT ACTY	BELVOIR
6M7F	04	0.00	MAB050	CS	CSS MPS	MACI COMML CONST EQUIP (CCE)	BELVOIR
				CS	EMW ENS	MACI HYDRAULIC SYS & COMP	BELVOIR
				CS	EMW ENS	CONSTRUCTION & COMBAT SUPPORT EQUIPMENT	BELVOIR
				CS	EMW ENS	MACI-HIGH SPEED EXCAVATOR	BELVOIR
6M7L	01	0.00	3T0046	CS	EMW ENS	PHY SEC EQ DEV & ACG	BELVOIR
			3T0048	CS	EMW ENS	ITEMS LESS THAN \$2.0M (CSS-EQ)	BELVOIR
			3T0049	CS	EMW ENS	MINES/COUNTERMINES	BELVOIR
			3T0051	CS	EMW ENS	ADVANCED MINE CLEARING SYS	BELVOIR
			3T0053	CS	EMW ENS	LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (LIMBS)	BELVOIR
			3T0056	CS	EMW ENS	IMPROVED LINE CHARGE	BELVOIR
			3T0057	CS	EMW ENS	IMPROVED HAND HELD MINE DETECTOR	BELVOIR
			3T0058	CS	EMW ENS	TACTICAL OBSTACLE MARKING SYSTEM (TOMS)	BELVOIR
			3T0307	CS	EMW ENS	REACTIVE FRAGMENT MINE NEUTRALIZATION SYSTEM (FRAG)	BELVOIR
			3T0003	CS	EMW ENS	WIDE-AREA NEUTRALIZATION MINE DEVICE (HAND)	BELVOIR
			3T0006	CS	EMW ENS	MAGNETIC SIGNATURE GILENCING SYSTEM (MASSIS)	BELVOIR
			3T0007	CS	EMW ENS	ELECTRIC EXPLOSIVE DETONATION SYSTEM	BELVOIR
			3T0028	CS	EMW ENS	AIRBORNE MINEFIELD DETECTION & RECONN SYS--AMTDARS	BELVOIR
			3T0037	CS	EMW ENS	SCATTERMINE RADAR (SCMR)	BELVOIR
			3T0038	CS	EMW ENS	MULTIPURPOSE DETECTION SYSTEM	BELVOIR
				CS	EMW ENS	CANETIP	BELVOIR
				CS	EMW ENS	AUTOMATED COUNTERBARRIER SYSTEM	BELVOIR
				CS	EMW ENS	VEHICLE HARDENING	BELVOIR

***** S A M P L E *****

Thu Aug 7 1986 11:58:30

pom89cum

This report presents the detailed pdip by pdip itemized listing of systems and workpackages uniquely linked to those systems. It gets its order from the pom89 data files.

expr [first pg#] ! acego ../neil/pom89cum [pg letter] [code] [name]

***** S A M P L E *****

MINES/COUNTERMINES (6M7L-01)
 POM BY PDIP-INCREMENT BREAKOUT FOR EMW
 WITH CUMULATIVE TOTALS (RTDE/PROC)

Acquisition Strategy:

SSN	TITLE	DA/TDC	MA	PDIP	87	88	89	90	91	92	93
3T0046	ADVANCED MINE CLEARING SYS			CS / EMW							
0	29-D 41-E 86-E 307-E 313-D										
	1. 63606 D60B 06 1124 BELVOIR Crit: 1										
	GENERIC TECHNOLOGY BASE - COUNTERMINE FLAITS										
	2. 63606 D60B 33 1307 BELVOIR Crit: 1								(548)		
	ADVANCED MECHANICAL MINE CLEARING DEVICE - INCREME										
	6M7L-01										
	Cumulative Totals: Procurement Funded:				0	0	0	0	0	0	0
	RDTE Funded:				0	0	0	0	0	0	0
3T0048	LIGHT INFANTRY MINEFIELD BREACHING SYSTE			CS / EMW							
0	29-C 41-B 86-D 149-B 313-D 327-A										
	1. 63606 D60B 07 1309 BELVOIR Crit: 1								440	2500	750
	LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (LIMBS)										
	2. 63606 D60B 07 1310 BELVOIR Crit: 1										
	LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (LIMBS)-								(2450)		
	6M7L-01										
	Cumulative Totals: Procurement Funded:				0	0	0	0	0	0	0
	RDTE Funded:				0	0	440	2500	0	750	0
3T0049	IMPROVED LINE CHARGE			CS / EMW							
0	29-C 41-B 86-D 149-B 313-D 327-B										
	1. 63619 D60B 08 01350 ARDEC Crit: 1										
	IMPROVED LINE CHARGE										
	6M7L-01										
	Cumulative Totals: Procurement Funded:				0	0	0	0	0	0	0
	RDTE Funded:				0	0	440	2500	0	750	0
3T0051	IMPROVED HAND HELD MINE DETECTOR			CS / EMW							
0	29-D 41-B 86-E 149-C 307-B 313-D 319-C										
	1. 63606 D60B 29 1049 BELVOIR Crit: 1										
	IMPROVED HAND HELD MINE DETECTOR										
	2. 63606 D60B 29 1300 BELVOIR Crit: 1								2800		800
	IMPROVED HAND HELD MINE DETECTOR - INCREMENT										
	3. 63606 D60B 49 1302 BELVOIR Crit: 2								(7800)	(9000)	(2500)
	ADVANCED TECHNOLOGY PORTABLE MINE DETECTOR - INCRE								(6600)	(8100)	(4600)
	6M7L-01										

***** S A M P L E *****

Thu Aug 7 1986 12:31:28

priorsys

C-60

This report produces an unclassified listing of the development systems for a single mission area in bdp priority order. All base case systems are placed at the top of the list.

expr [first pg#] ; acego ../neil/priorsys [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

CSS I-N DEVELOPMENT SYSTEM BDP RANKING

SSN	DA MA	AMC MGR	TITLE	#DEF	#A	#B	#C	#D	#E	#X	BL	RATING
1. K51600	CSS	PM-TMDE	CONTACT TEST SET	8	1	2	5	2	0	0	0	base case
2. 082500	CSS	TACOM	RECOVERY VEHICLE, MED, FT, MB8A1, MOD	7	1	1	2	1	0	0	0	base case
3. M87001	CSS	PM-TMDE	TMDE FOR STE/ICE	6	0	5	1	0	0	0	0	base case
4. M87003	CSS	PM-TMDE	CORE ELECTRONIC AUTO TEST (STE-X)	3	0	5	0	0	0	0	0	base case
5. M48800	CSS	BELVOIR	TRUCK, FORK LIFT, DE, FT, RT, 10000 LB	3	1	0	0	2	0	0	0	base case
6. M11200	CSS	BELVOIR	LOGISTIC SUPPORT VESSEL (LSV)	2	0	2	0	0	0	0	0	base case
7. M34200	CSS	BELVOIR	LANDING, CRAFT, UTILITY	2	0	2	0	0	0	0	0	base case
8. D14031	CSS	TACOM	TRUCK, BOLSTER, 5 TON, 6x6	4	0	0	3	0	1	0	0	base case
9. M61900	CSS	AMCCOM	SHOP EG CONTACT MAINTENACE TRK MTD (MVP)	2	0	1	1	0	0	0	0	base case
10. M61900	CSS	AMCCOM	SHOP EQUIPMENT, ORG REPAIR, TRK MTD	2	0	1	1	0	0	0	0	base case
11. D06800	CSS	TACOM	TRAILER TANK WATER 400G 1 1/2T 2M M149A1	2	0	1	1	0	0	0	0	base case
12. MA3120	CSS	PM-PHL	SWA PETROLEUM DISTRIBUTION SYSTEM	1	0	1	0	0	0	0	0	base case
13. MA4021	CSS	PM-PHL	WATER PURIFICATION UNIT, REV OSMOSIS, 600 GPH	2	0	1	1	0	0	0	0	base case
14. M87600	CSS	NATICK	LAUNDRY UNIT/TRL MTD	1	1	0	0	0	0	0	0	base case
15. R09900	CSS	BELVOIR	CAUSEWAY, FLOATING	4	0	0	2	0	0	0	0	base case
16. M34300	CSS	BELVOIR	AIR CUSHION VEHICLE, 29-30 TON	2	0	0	2	0	0	0	0	base case
17. M62700	CSS	AMCCOM	WELDING SHOP, TRAILER MTD	2	0	0	2	0	0	0	0	base case
18. M19000	CSS	PM-PHL	TANK ASSEMBLY FAB COL-L POL 50000 G	1	0	1	0	0	0	0	0	base case
19. M27400	CSS	PH-PHL	TACTICAL WATER DISTR SYS	1	0	1	0	0	0	0	0	base case
20. M86600	CSS	NATICK	FIELD KITCHEN, MOBILE, TRL MTD	1	0	1	0	0	0	0	0	base case
21. R18800	CSS	PM-PHL	ARCTIC FORWARD AREA REFUELING	1	0	0	1	0	0	0	0	base case
22. R18900	CSS	PM-PHL	ARCTIC FUEL SYSTEM SUPPLY POINT	1	0	0	1	0	0	0	0	base case
23. MA1100	CSS	PM-AMHLOG	LOG UNIT PROD-UNIT PROD/TRANS	1	0	0	1	0	0	0	0	base case
24. R21400	CSS	BELVOIR	TANK ASSY, FAB COLLAPS, 20,000 GAL POL	1	0	1	0	0	0	0	0	base case
25. A39900	CSS	NATICK	TRANSPORTABLE HELICOPTER ENCLOSURE (THE)	0	0	0	0	0	0	0	0	base case
26. M03400	CSS	NATICK	HALON RECHARGE SERVICE KIT	0	0	0	0	0	0	0	0	base case
27. M47400	CSS	BELVOIR	TRUCK, FORK LIFT, GE, PT, 4000 LB	0	0	0	0	0	0	0	0	base case
28. M62300	CSS	PM-PHL	TANK/PUMP UNIT LIG DISP F/TRK MOUNTING	0	0	0	0	0	0	0	0	base case
29. MA5175	CSS	TACOM	HOST NATION SUPPORT - EUROPE	0	0	0	0	0	0	0	0	base case
30. M32000	CC	TACOM	ARMORED FAMILY OF VEHICLES (AFV)	6	4	2	0	0	0	0	0	3560.00
31. M03304	CSS	TACOM	CBT LOG VEH	15	2	3	4	5	1	0	0	3071.80
32. 5L0003	CSS	PM-AMLOG	AMMUNITION LOGISTICS	15	2	3	3	2	2	0	0	2937.50
33. M2019	CSS	TACOM	AFV - AMMO RESUPPLY (REARM)	12	2	3	3	2	2	0	0	2835.20
34. M2020	CSS	TACOM	AFV - REFUEL VEHICLE	11	2	3	2	3	1	0	0	2813.00
35. 6C0001	CSS	PM-TMDE	DIAGNOSTICS, PROGNOSTICS, AND ATE TECHNOLOGY	9	2	5	3	1	0	0	0	2694.20
36. M2023	CSS	TACOM	AFV-RECOVERY VEHICLE (MVB FOLLOW-ON)	12	1	3	3	5	0	0	0	2549.60
37. M2001	CSS	TACOM	ARMORED MAINTENANCE VEHICLE (NEAR TERM)	13	1	3	2	7	0	0	0	2497.20
38. M2022	CC	TACOM	AFV/ARMORED MAINTENANCE VEHICLE (AMV)	8	1	3	3	1	0	0	0	2352.20
39. 4T5334	CSS	NATICK	CLOTHING/LIFE SUPPORT EQUIP, INTEGRATED, CB PROT	13	1	3	1	2	6	0	0	2230.40
40. 3R0102	CSS	MRDEC	MEDICAL DEVELOPMENT SYSTEM	13	1	3	1	2	6	0	0	2238.40
41. 6R0101	CSS	MRDEC	MEDICAL TECH BASE SYSTEM	10	1	3	5	1	0	0	0	2121.80
42. 8Z5270	CSS	CECOM	TMDE MODERNIZATION	10	1	3	6	0	0	0	0	2131.60
43. K18400	CSS	PM-TMDE	BASE SHOP TEST FACILITY	8	2	2	3	1	0	0	0	2116.70
44. 3C0003	CSS	PM-TMDE	TEST PROGRAM SET (TPS) STANDARDS & TOOLS	9	2	1	1	3	2	0	0	1987.90
45. M2030	CSS	TACOM	AFV-BATTALION AID STATION (BAS AID)	9	2	1	1	3	2	0	0	1987.90
46. M2021	CSS	TACOM	AFV-MEDICAL EVACUATION VEHICLE	16	0	2	6	7	1	0	0	1953.60
47. 6L0002	CSS	PM-AMLOG	AMMO PACKAGING TECHNOLOGY DEMO									

***** U N C L A S S I F I E D *****

Thu Jul 17 1986 18:00:09

priority

This report produces a workpackage 1 to N list similar to priority, except that only tech base (6.1, 6.2, and 6.3A) workpackages are shown. The workpackages are those linked to systems in the system control file.

expr [first pg#] | acego ../neil/priority [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

BDP TECH BASE WORKPACKAGE RANKINGS
FOR EMH

PE	PROJ TASK	NO	TITLE	COMMAND	RATING	# SYS
1.	62601 AH91 01	ZE110	LOGISTICAL SUPPORT TECHNOLOGY	TACOM	180323.91	10
2.	62601 AH91 35	R55001	VETRONICS TECHNICAL AREA	TACOM	93772.45	14
3.	63631 D424 67	RC58500	ADVANCED TRACK & SUSPENSION TECHNOLOGY	TACOM	83459.97	11
4.	62601 AH91 58	R55002	VETRONICS/(INT)-2/BATTLEFIELD MANAGEMENT SYSTEM	TACOM	77724.20	14
5.	63631 D014 62	R55401	BATTLEFIELD DIAGNOSTIC/PROGNOSTIC TECHNOLOGY	TACOM	71995.23	12
6.	63621 D007 69	R5771	LIGHT MOBILITY PLATFORM - A (SIPS)	TACOM	69394.98	8
7.	62601 AH91 65	RC56600	TRACK & SUSPENSION TECHNOLOGY	TACOM	64322.71	14
8.	63631 D424 67	RC37900	LIGHT MOBILITY PLATFORM - C (TRACK & SUSPENSION)	TACOM	63232.73	11
9.	63631 D424 67	RC37901	LIGHT MOBILITY PLATFORM - C (TRACK & SUSPENSION) (UNFUNDED)	TACOM	61229.98	11
10.	63631 D424 67	RC37500	MEDIUM MOBILITY PLATFORM - C (TRACK & SUSPENSION)	TACOM	57535.98	9
11.	63621 D607 69	R378	MOBILITY INTEGRATION A.3	TACOM	54704.48	11
12.	62601 AH91 43	403	LIFE SUPPORT SYSTEMS TECHNOLOGY	TACOM	53563.98	9
13.	62601 AH91 43	403	LIFE SUPPORT SYSTEMS TECHNOLOGY	TACOM	53563.98	9
14.	62733 AH20 V	5021	MOBILITY FUELS, LUBRICANTS & DETERIORATION PREVENTION	BELVOIR	48404.22	33
15.	63631 D014 57	NK09	V(INT)-2/BATTLEFIELD MANAGEMENT SYSTEM	TACOM	44241.73	13
16.	63631 D014 57	R55201	V(INT)-2/BATTLEFIELD MANAGEMENT SYSTEM	TACOM	43590.24	13
17.	63606 D608 16	1023	WIDE AREA NEUTRALIZATION DEVICE (HAND)	BELVOIR	43012.00	2
18.	63606 D608 08	1006	COUNTEROBSTACLE VEHICLE	BELVOIR	42195.50	3
19.	62733 AH20 01	1003	COUNTERMINE EXPLORATORY DEVELOPMENT	BELVOIR	41753.99	4
20.	63631 D014 59	R53102	VETRONICS TECHNOLOGY AREA	TACOM	39131.24	9
21.	63631 D014 44	404	LIFE SUPPORT SYSTEMS DEMONSTRATION	TACOM	39077.49	9
22.	61102 AH31 F	5020	MOBILITY FUELS, LUBRICANTS & CORROSION RESEARCH	BELVOIR	37282.48	25
23.	63606 D608 01	1034	BARRIER 21/ANTI-TANK OBSTACLE	BELVOIR	37294.99	3
24.	63640 D543 A1	033L3	UNIVERSAL TANK AMMUNITION PACKAGING (GALS)	ARDEC	36409.99	4
25.	63640 D543 A1	033L7	AMMO PACKAGING/HANDLING SYSTEM (APHS) FOR CORVAT	ARDEC	35600.00	1
26.	62105 AH84 03	3207	CHEMICAL PROTECTION	LABCOM	35594.49	10
27.	62601 AH91 26	RR150	ADVANCED GROUND VEHICLE TECHNOLOGY (AGVT) FOLLOW-ON	TACOM	33637.99	5
28.	63606 D608 11	1007	AIRBORNE MINEFIELD DETECTION & RECONNAISSANCE SYSTEM	BELVOIR	33320.00	1
29.	63606 D608 48	1315	MINEFIELD IMAGE RECOGNITION & CUEING STATIONS (MIRACS)	BELVOIR	33320.00	1
30.	63606 D608 05	1005	CANE TIP	BELVOIR	31402.00	2
31.	63606 D608 27	1340	UNFINANCED COUNTERMINE PROGRAM	BELVOIR	29792.99	5
32.	63606 D608 27	1341	UNFINANCED COUNTERMINE PROGRAM - INCREMENT	BELVOIR	29792.99	5
33.	63631 D014 61	R55301	SECURED LIGHTING	TACOM	29319.74	14
34.	63606 D608 30	1012	SCATTERABLE MINEFIELD DETECTION SYSTEM (SCH)	BELVOIR	29277.00	3
35.	62601 AH91 26	RR120	COMPUTER AIDED REMOTE DRIVING (CARD)	TACOM	26922.50	5
36.	62601 AH91 26	RR140	ROBOTIC VEHICLE VISION TECHNOLOGY	TACOM	26922.50	5
37.	62601 AH91 26	RR180	ROBOTIC VEHICLE COMMUNICATION TECHNOLOGY	TACOM	26922.50	5
38.	63104 D150 01	5004	FIRE-RESISTANT FUEL TESTING WITH EQUIPMENT	BELVOIR	26445.49	23
39.	63102 D01 01	1047	COMBAT ENGINEER COMPONENTS	BELVOIR	25410.99	5
40.	63606 D608 02	1079	MAN-MADE OBSTACLE REPORTING AND DISPOSITION SYSTEM (MORDS)	BELVOIR	25075.00	3
41.	63640 D543 A1	033L4	GENERIC LP REBUPPLY VEHICLE	ARDEC	24259.00	2
42.	63606 D608 25	1009	IMPROVED LINE CHARGE	BELVOIR	23649.00	2
43.	63104 D150 06	5005	FIELD FUEL QUALITY MONITOR TEST	BELVOIR	22529.49	25
44.	62705 AH94 11	5019	THERMOELECTRIC GENERATOR - IPECS	LABCOM	22260.00	2
45.	63104 D150 09	5008	FUEL PRETREATMENT UNIT	BELVOIR	21604.49	24
46.	62601 AH91 35	IS271	MILITARY VEHICLE DESIGN/PERFORMANCE OPTIMIZATION	TACOM	20879.99	10

***** U N C L A S S I F I E D *****

Tue Jul 29 1986 11:36:48

priorwp

This report produces an unclassified listing of the workpackages for a single mission area in bdp priority order. (6.3, 6.4, and 6.7)

expr [first pg#] | acego ../neil/priorwp [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

CSS I-N WORKPACKAGE BDP RANKINGS

PE	PROJ TASK	NO	TITLE	COMMAND	RATING	# SYS
1.	63640 D543 A1	00306	BATTLEFIELD RESUPPLY	ARDEC	48290.24	9
2.	63209 D833 I0	RTL3078	IN-HOUSE CARGO HANDLING EQUIPMENT	AVSCOM	43941.98	4
3.	63209 D832 04	RTL3076	IN-HOUSE GROUND SUPPORT EQUIPMENT	AVSCOM	42569.99	5
4.	63640 D543 A1	033L3	UNIVERSAL TANK AMMUNITION PACKAGING (GALS)	ARDEC	36044.75	4
5.	63640 D543 A1	033L7	AMMO PACKAGING/HANDLING SYSTEM (APHS) FOR COMBAT	ARDEC	35600.00	1
6.	63747 D669 8V	D669-8V	POL RESISTANT CHEM PROT TACTILE GLOVE	NATICK	33808.00	2
7.	64713 DL40 0Y	DL40-0Y	INDIVIDUAL SOLDIER MICROCLIMATE COOLING SYSTEM	NATICK	33808.00	2
8.	64713 DL40 1Z	DL40-1Z	POL RESISTANT CHEMICAL PROTECTIVE GLOVE	NATICK	33372.00	2
9.	64717 DL41 16	2042	IMPROVED REFUELING SYSTEM	PM-PWL	33372.00	2
10.	63747 D669 3Z	D669-3Z	INTEGRATED PROTECTIVE CLOTHING SYSTEM FOR GROUND TROOPS	NATICK	32110.00	2
11.	63747 D669 40	D669-40	OVERBOOT, MULTI-PURPOSE, RAIN/SNOW/CP (MULO)	NATICK	32110.00	2
12.	63640 D543 A1	00309	ROBOTIC AMMUNITION RESUPPLY VEHICLE (RARV)	ARDEC	30046.50	2
13.	64746 DL59 01	L590101	IFTE TPS IN HOUSE	CECOM	28626.00	2
14.	64746 DL59 01	L590102	IFTE TPS FSED	CECOM	28626.00	2
15.	64746 DL59 01	L590103	IFTE TPS PROGRAM SUPPORT	CECOM	28626.00	2
16.	64746 DL59 01	L590104	IFTE TPS	CECOM	28626.00	2
17.	63748 D28 01	J280103	ATSE LCSE SUPPORT (I)	CECOM	28130.00	1
18.	63748 D28 02	J280202	ATSE NETWORKING	CECOM	28130.00	1
19.	63748 D28 02	J280203	ATSE NETWORKING	CECOM	28130.00	1
20.	63748 D28 03	J280301	PIN ELECTRONICS	CECOM	28130.00	1
21.	63748 D28 03	J280302	INFRARED DIAGNOSTICS	CECOM	28130.00	1
22.	63748 D28 03	J280305	SPREAD SPECTRUM TESTING TECHNOLOGY	CECOM	28130.00	1
23.	63748 D28 04	J280402	BIT REQUIREMENTS ANALYSIS MODEL (BRAN)	CECOM	28130.00	1
24.	63748 D28 04	J280403	TEST SUPPORT ANALYSIS MODEL (TSAM) II, II & IV	CECOM	28130.00	1
25.	63747 D669 1Y	D669-1Y	INDIVIDUAL SOLDIER MICROCLIMATE COOLING SYSTEM	NATICK	27816.00	2
26.	64713 DL40 1V	DL40-1V	LIGHT FIGHTER CHEMICAL PROTECTIVE ENSEMBLE (LITE PROTECTOR)	NATICK	27090.00	2
27.	64713 DL40 1V	DL40U1V	LIGHT FIGHTER CHEMICAL PROTECTIVE ENSEMBLE (LITE PROTECTOR)	NATICK	27090.00	2
28.	63104 D150 01	3004	FIRE-RESISTANT FUEL TESTING WITH EQUIPMENT	BELVOIR	26445.49	23
29.	63747 D669 7W	D669-7W	LIGHT FIGHTER CHEMICAL PROTECTIVE ENSEMBLE (LITE PROTECTOR)	NATICK	25306.00	2
30.	63640 D543 A1	033L4	GENERIC LP RESUPPLY VEHICLE	ARDEC	24259.00	2
31.	63747 D669 3S	D669-3S	GROUND/AIR MICROCLIMATE COOLING SYSTEM	NATICK	23522.00	1
32.	63747 D669 6Z	D669-6Z	INTEGRATED COMBAT VEHICLE CREWMAN CLOTHING SYSTEM	NATICK	23522.00	1
33.	63747 D669 7U	D669-7U	TOXIC AGENT AND CHEMICAL HANDLER SYSTEM	NATICK	23522.00	1
34.	63747 D669 8V	D669-8V	POL RESISTANT CHEM PROT TACTILE GLOVE	NATICK	23522.00	1
35.	63747 D669 8W	D669-8W	INTEGRATED LIGHTWEIGHT COMBAT BOOT	NATICK	23522.00	1
36.	63747 D669 8W	D669-8W	INTEGRATED LIGHTWEIGHT COMBAT BOOT	NATICK	23522.00	1
37.	63747 D669 8Y	D669-8Y	IMPERMEABLE CHEMICAL PROTECTIVE SUIT	NATICK	23522.00	1
38.	63747 D669 9V	D669-9V	ADVANCED INTEGRATED COMBAT UNIFORM FOR GROUND TROOPS	NATICK	23522.00	1
39.	63747 D669 9W	D669-9W	ADVANCED INTEGRATED HANDWEAR, CW/BALLISTIC PROTECTIVE	NATICK	23522.00	1
40.	63747 D669 9X	D669-9X	MANEUVER ARM TACTICAL PROTECTIVE SYSTEM	NATICK	23522.00	1
41.	63747 D669 9Z	D669-9Z	INTEGRATED (CB PROT) CVC HELMET SYSTEM	NATICK	23522.00	1
42.	64713 DL40 0W	DL40-0W	INTEGRATED COMBAT VEHICLE CREWMAN CLOTHING SYSTEM	NATICK	23522.00	1
43.	64713 DL40 0W	DL40-0W	INTEGRATED COMBAT VEHICLE CREWMAN CLOTHING SYSTEM	NATICK	23522.00	1
44.	64713 DL40 13	DL40-13	GLOVE, CHEM PROT, LIGHTWEIGHT, TACTILE	NATICK	23522.00	1
45.	64713 DL40 1U	DL40-1U	CLOTHING, PETROLEUM PROTECTIVE, EXTREME COLD WEATHER	NATICK	23522.00	1
46.	64713 DL40 1Z	DL40U1Z	POL RESISTANT CHEMICAL PROTECTIVE GLOVE	NATICK	23522.00	1
47.	64713 DL40 36	DL40-36	AIRCREW UNIFORM INTEGRATED BATTLEFIELD (AUIB)	NATICK	23522.00	1

***** U N C L A S S I F I E D *****

Thu Jul 17 1986 18:03:20

prisysda2

This report produces an unclassified decision aid of systems in 1 to N BDP order. It includes the uniquely linked workpackages under each system.

expr [first pg#] i acego ../neil/prisysda2 [pg letter] [code] [name]

***** UNCLASSIFIED *****

BDP SYSTEM RANKING DECISION AID FOR EMW
WITH CUMULATIVE TOTALS (RTDE) (continued)

SSN	TITLE	DA/TDC MA	BDP	87	88	89	90	91	92	93
370301	ADVANCED GENERATOR SET COMPONENTS 1. 63702 D011 26 2202 BELVOIR Crit: 1 ADVANCED GENERATOR SET COMPONENTS 2. 63702 D011 26 2206 BELVOIR Crit: 1 ADVANCED GENERATOR SET COMPONENTS - INCREMENT Cumulative Totals: RDTE Funded: RDTE Unfunded:	CS / EMW	625	80921 58787 64488 66032 71287 80998 106195 28277 43800 24693 41816 43171 33668 1900					800	800
370007	MULTIPURPOSE DETECTION SYSTEM 1. 63619 D606 37 11331 ARDEC Crit: 1 MULTI-PURPOSE DETECTION SYSTEM (RISK) 2. 63619 D606 23 11353 ARDEC Crit: 1 UNIVERSAL PORTABLE MINE DETECTOR (UPMD) (RISK) 3. 63606 D608 36 1081 BELVOIR Crit: 1 MULTIPURPOSE DETECTION SYSTEM 4. 63606 D608 36 1308 BELVOIR Crit: 1 MULTIPURPOSE DETECTION SYSTEM - INCREMENT Cumulative Totals: RDTE Funded: RDTE Unfunded:	CS / EMW	607		1213 2249	991 400	2400	(2000)		
370028	CANETIP 1. 64612 D415 29 01347 ARDEC Crit: 1 CANE TIP (MIN) 2. 64612 D415 29 11347 ARDEC Crit: 1 CANE TIP (RISK) 3. 64612 D415 29 91347 ARDEC Crit: 1 CANE TIP Cumulative Totals: RDTE Funded: RDTE Unfunded:	CS / EMW	607	80921 58787 65125 68765 74523 83698 106195 28277 43800 24693 41816 45171 33668 1900		3238 3605 6234		(2300)		6272
E37401	PENETRATION AUGMENTATION MUNITION (PAM) 1. 64612 D021 04 01328 ARDEC Crit: 1 PENETRATION AUGMENTED MUNITION (PAM) ED EFFORT 2. 63619 D607 01 01314 ARDEC Crit: 1 PENETRATION AUGMENTED MUNITION (PAM) AD EFFORT Cumulative Totals: RDTE Funded: RDTE Unfunded:	CS / SOF	593	80921 58787 65125 72003 78128 89932 112467 28277 43800 24693 41816 47671 33668 1900		1950 4100 3300			1407 1610	

***** UNCLASSIFIED *****

Fri Jul 25 1986 13:47:35

procpri

This report produces an unclassified or confidential listing of the systems under procurement (type classified systems) for a single mission area. The systems are listed in PDIP increment priority, and if a system appears in more than one increment, each one is listed separately.

expr [first pg#] i acego ../neil/procpri [pg letter] [code] [name] [class]

***** U N C L A S S I F I E D *****

DA PROCUREMENT PRIORITY FOR SOF

SSN	DA MA	AMC MSC	TITLE	PDJP	PRIORITY
1.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	FL7R-01	
2.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	FL7R-02	
3.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6XC2-01	
4.	BA950A	C3	INIT SP/RP PARTS-TELECOM	FL6N-01	
5.	BA950A	C3	INIT SP/RP PARTS-TELECOM	FL7J-01	
6.	BA950A	C3	INIT SP/RP PARTS-TELECOM	5P7H-01	
7.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6M74-01	
8.	B74402	C4	LIGHTWEIGHT DIGITAL FACSIMILE	6XG5-01	
9.	BA950A	C3	INIT SP/RP PARTS-TELECOM	6XG5-01	
10.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6R8B-01	
11.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	FL7H-01	
12.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6M7H-01	
13.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6V3H-01	
14.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6S72-01	
15.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6S7E-01	
16.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6X4V-01	
17.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	FL7X-01	
18.	BA950A	C3	INIT SP/RP PARTS-TELECOM	5S7H-01	
19.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6XLB-01	
20.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	3R7H-01	
21.	BA950A	C3	INIT SP/RP PARTS-TELECOM	6S5X-01	
22.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6S5X-01	
23.	BA950A	C3	INIT SP/RP PARTS-TELECOM	6S4K-01	
24.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6X3V-01	
25.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	FL7P-01	
26.	BL5292	C3	ITEMS LESS THAN \$2.0M (EM-C-E)	FL7P-01	
27.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6S8V-01	
28.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6X9X-01	
29.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6XE1-01	
30.	BA950A	C3	INIT SP/RP PARTS-TELECOM	6S5U-01	
31.	B10500	C3	PWR SUPPLY PP-6148/U	CP6V-01	
32.	B74402	C4	LIGHTWEIGHT DIGITAL FACSIMILE	CP6V-01	
33.	BA950A	C3	INIT SP/RP PARTS-TELECOM	CP6V-01	
34.	BL3220	C3	ITEMS LESS THAN \$2.0M (CSC-C-E)	CP6V-01	
35.	BA950A	C3	INIT SP/RP PARTS-TELECOM	6B01-01	
36.	4M529F	C4	TACTICAL SATELLITE ECM	6S01-01	
37.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	2SA3-01	
38.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	2SB0-01	
39.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	FL7L-01	
40.	BA950A	C3	INIT SP/RP PARTS-TELECOM	6S5H-01	
41.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	CP7C-01	
42.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	1S6N-01	
43.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6P31-01	
44.	BA950A	C3	INIT SP/RP PARTS-TELECOM	5S4T-01	
45.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6W3C-01	
46.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6R8J-01	
47.	BA960A	C3	INIT SP/RP PARTS-OTHER (C-E)	6R8L-01	

***** U N C L A S S I F I E D *****

projects

This diagnostic report shows the contents of the project control file.

acego ../qa/projects

COMMAND	PE	PROJ	CAT	SUBCAT	CODE	NAME	TITLE
ARDEC	23751	D063	6.3	6.7	1	SOF	SOF FIRE CONTROL DEV
ARDEC	62623	AH21	6.2	6.2	1	SOF	JOINT SERVICES SMALL ARMS PROGRAM
ARDEC	62624	AH28	6.2	6.2	1	EMW	MUNITIONS TECHNOLOGY
ARDEC	63606	D006	6.3	6.3A	1	EMW	LANDMINE WARFARE DEVELOPMENT
ARDEC	63607	D627	6.3	6.3A	1	SOF	JOINT SERVICES ARMS PROGRAM (JSSAP)
ARDEC	63608	D332	6.3	6.3B	1	SOF	INDIVIDUAL SOLDIERS WEAPONS
ARDEC	63619	D005	6.3	6.3B	1	EMW	LANDMINE SYSTEMS
ARDEC	63619	D606	6.3	6.3B	1	EMW	COUNTERMINES AND BARRIERS
ARDEC	63619	D607	6.3	6.3B	1	EMW	EXPLOSIVE DEMOLITION SYSTEMS
ARDEC	63619	D607	6.3	6.3B	1	SOF	EXPLOSIVE DEMOLITION SYSTEMS
ARDEC	63619	DL01	6.3	6.3B	1	EMW	COUNTER OBSTACLE VEHICLE
ARDEC	63640	D943	6.3	6.3A	1	CSS	AMMUNITION LOGISTICS
ARDEC	64612	D021	6.3	6.4	1	EMW	EXPLOSIVE DEMOLITIONS
ARDEC	64612	D021	6.3	6.4	1	SOF	EXPLOSIVE DEMOLITIONS
ARDEC	64619	D415	6.3	6.4	1	EMW	MINE DETECTION AND NEUTRALIZATION
ARDEC	64619	D016	6.3	6.4	1	EMW	MINE SYSTEMS
ARDEC	64XXX	DTTT	6.3	6.4	1	CSS	TOOL SETS KITS AND OUTFITS
ARDEC	69810	DE63	6.3	6.5	1	CSS	MIL ADAPT COM TECH
ARDEC	69810	DE63	6.3	6.5	1	EMW	MIL ADAPT COM TECH
AVSCOM	23751	D615	6.3	6.7	1	SOF	SOF AVIATION
AVSCOM	63209	D832	6.3	6.3B	1	CSS	GROUND SUPPORT EQUIPMENT
AVSCOM	63209	D833	6.3	6.3B	1	CSS	CARGO HANDLING EQUIPMENT
AVSCOM	64204	DC32	6.3	6.4	1	CSS	GROUND SUPPORT EQUIPMENT
AVSCOM	64204	DC33	6.3	6.4	1	CSS	CARGO HANDLING EQUIPMENT
AVSCOM	69810	DE65	6.3	6.5	1	CSS	MIL ADAPT COMP TECH
BELVOIR	61102	AH51	6.1	6.1	3	BELVOIR	COMBAT SUPPORT
BELVOIR	61102	AH51	6.1	6.1	1	CSS	COMBAT SUPPORT
BELVOIR	62733	AH20	6.1	6.1	1	EMW	COMBAT SUPPORT
BELVOIR	62733	AH20	6.2	6.2	3	BELVOIR	MOBILITY EQUIPMENT TECHNOLOGY
BELVOIR	62733	AH20	6.2	6.2	1	CSS	MOBILITY EQUIPMENT TECHNOLOGY
BELVOIR	62733	AH20	6.2	6.2	1	EMW	MOBILITY EQUIPMENT TECHNOLOGY
BELVOIR	63102	DJ01	6.3	6.3A	3	BELVOIR	COMBAT ENG COMPONENTS
BELVOIR	63102	DJ01	6.3	6.3A	1	EMW	COMBAT ENG COMPONENTS
BELVOIR	63104	D150	6.3	6.3A	3	BELVOIR	FUELS AND EQUIPMENT
BELVOIR	63104	D150	6.3	6.3A	1	CSS	FUELS AND EQUIPMENT
BELVOIR	65604	D608	6.3	6.3A	3	BELVOIR	COUNTERMINE AND BARRIER DEVELOPMENT
BELVOIR	63606	D608	6.3	6.3A	1	EMW	COUNTERMINE AND BARRIER DEVELOPMENT
BELVOIR	63702	D611	6.3	6.3B	3	BELVOIR	ADVANCED ELECTRICAL ENERGY CONCEPTS
BELVOIR	63702	D611	6.3	6.3B	1	EMW	ADVANCED ELECTRICAL ENERGY CONCEPTS
BELVOIR	63712	D580	6.3	6.3B	3	BELVOIR	FIELD ARMY MAPPING
BELVOIR	63712	D580	6.3	6.3B	1	EMW	FIELD ARMY MAPPING
BELVOIR	63712	DT44	6.3	6.3B	3	BELVOIR	DIGITAL TOPOGRAPHIC SUPPORT SYSTEM
BELVOIR	63712	DT44	6.3	6.3B	1	EMW	DIGITAL TOPOGRAPHIC SUPPORT SYSTEM
BELVOIR	63726	D471	6.3	6.3B	3	BELVOIR	COUNTER SURVEILLANCE AND TACTICAL DECEPTION
BELVOIR	63726	D901	6.3	6.3B	3	BELVOIR	GENERAL SUPPORT EQUIPMENT
BELVOIR	63726	D901	6.3	6.3B	1	EMW	GENERAL SUPPORT EQUIPMENT
BELVOIR	63726	D014	6.3	6.3B	3	BELVOIR	CONTAINER DISTR EQUIP
BELVOIR	63726	D014	6.3	6.3B	1	CSS	CONTAINER DISTR EQUIP
BELVOIR	63726	DK39	6.3	6.3B	3	BELVOIR	GENERAL SPT EQUIPMENT
BELVOIR	63726	DK39	6.3	6.3B	1	CSS	GENERAL SPT EQUIPMENT
BELVOIR	63726	DK39	6.3	6.3B	1	EMW	GENERAL SPT EQUIPMENT
BELVOIR	63755	D605	6.3	6.3B	1	IEW	BATTLEFIELD SUSTAINMENT
BELVOIR	63773	D124	6.3	6.3A	1	CSS	BATTLEFIELD SUSTAINMENT
BELVOIR	638XX	DXXX	6.3	6.3B	3	BELVOIR	BATTLEFIELD SUSTAINMENT
BELVOIR	63LIC	DLIC	6.3	6.3A	3	BELVOIR	BATTLEFIELD SUSTAINMENT
BELVOIR	63LIC	DLIC	6.3	6.3A	1	SOF	BATTLEFIELD SUSTAINMENT
BELVOIR	63XXX	DX33	6.3	6.3A	3	BELVOIR	BATTLEFIELD SUSTAINMENT

LOW INTENSITY CONFLICT - NEW PROPOSED 6.3A PROGRAM
TACTICAL DECEPTION AND COUNTERSURVEILLANCE SYS

projindex

This report produces an unclassified listing of the projects in the project control file, for a specific mission area, as an index to the projwkp listing.

expr [first pg#] | acego ../neil/projindex [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

CSS FUNDED RDTE PROJECT INDEX

PE	PROJ	TITLE	COMMAND	PAGE
23735	D280	RECOVERY VEHICLE IMPROVEMENT PROGRAM	TACOM	----
23754	D420	P31 PROGRAM	CECOM	----
61102	A318	NIGHT VISION AND ELECTRO OPTIC RESEARCH	CECOM	----
61102	AF22	RESEARCH IN VEHICLE MOBILITY	TACOM	----
61102	AH51	COMBAT SUPPORT	BELVOIR	----
61102	AH52	EQUIPMENT FOR THE INDIVIDUAL SOLDIER	NATICK	----
62105	AH84	MATERIALS	LABCOM	----
62111	AH71	ATMOSPHERIC INVESTIGATIONS	LABCOM	----
62210	D283	AIRDROP TECHNOLOGY	LABCOM	----
62601	AH91	TANK-AUTOMOTIVE TECHNOLOGY	NATICK	----
62703	AH94	ELECTRONICS & ELECTRON DEVICES	TACOM	----
62723	A427	TACTICAL RIGID WALL SHELTER EXPLORATORY	LABCOM	----
62723	AH98	CLOTHING AND EQUIPMENT TECHNOLOGY	NATICK	----
62724	AH99	JOINT SERVICES FOOD/NUTRITION TECHNOLOGY	NATICK	----
62733	AH20	MOBILITY EQUIPMENT TECHNOLOGY	NATICK	----
63104	D150	FUELS AND EQUIPMENT	BELVOIR	----
63209	D832	GROUND SUPPORT EQUIPMENT	BELVOIR	----
63209	D833	CARGO HANDLING EQUIPMENT	AVSCOM	----
63218	D242	AIRDROP TECHNOLOGY DEMONSTRATION	AVSCOM	----
63223	D266	AIRDROP EQUIPMENT AND TECHNOLOGY	NATICK	----
63624	DH06	VEHICLE MOBILITY ENHANCEMENT	NATICK	----
63640	D543	AMMUNITION LOGISTICS	TACOM	----
63703	DM62	PHYSICAL SECURITY	ARDEC	----
63726	D428	TACTICAL RIGID WALL SHELTER ADVANCED DEV	PHPSF	----
63726	D526	MARINE ORIENTED LOG EQ	NATICK	----
63726	DG14	CONTAINER DISTR EQUIP	PH-AHC	----
63726	DK39	GENERAL SPT EQUIPMENT	BELVOIR	----
63726	DK39	GENERAL SUPPORT EQUIPMENT	BELVOIR	----
63726	DK41	POL DISTRIBUTION SYS	PH-PWL	----
63747	D610	FOOD ADVANCED DEVELOPMENT	PH-PWL	----
63747	D669	CLOTHING AND EQUIPMENT	NATICK	----
63748	D244	TPS STANDARDS & TOOLS	NATICK	----
63748	D594	METROLOGY & CALIB	CECOM	----
63748	DJ28	DIAGNOSTIC AND STANDARD ATE	HICOM	----
63773	D124	BATTLEFIELD SUSTAINMENT	CECOM	----
63899	D064	RDTE 91-2001 WEDGE	TACOM	----
63999	D041	GROUND SUPPORT EQUIPMENT	HO-AHC	----
64204	DC32	CARGO HANDLING EQUIPMENT	AVSCOM	----
64204	DC33	AIRDROP EQUIPMENT DEVELOPMENT	AVSCOM	----
64218	D279	FAMILY OF HEAVY TACTICAL VEHICLES (FHTV)	NATICK	----
64604	D659	FAMILY OF MEDIUM TACTICAL VEHICLES	TACOM	----
64604	DH07	MILITARY SUBSISTENCE SYSTEMS	TACOM	----
64713	D548	CLOTHING AND EQUIPMENT	NATICK	----
64713	DL40		NATICK	----

***** U N C L A S S I F I E D *****

projwkp

This report produces a project-workpackage table, including the workpackage funding and the systems linked to each workpackage.

expr [first pg#] : acego ../neil/projwkp [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

PROJECT/WORKPACKAGE SUMMARY

COMMAND: ARDEC PROJECT: 63606 D006 TITLE: LANDMINE WARFARE DEVELOPMENT

TASK/WKP/SSN	WORKPACKAGE/SYSTEM TITLE	PDIP	FY86	FY87	FY88	FY89	FY90	FY91	FY92	FY93
01	MINE SENSORS	-								
E71200	MODULAR PACK MINE SYSTEM	2*								
E72195	CANISTER MINE (VOLCAND) XMB7	2*								
WB1038	IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)	1*								
01	MINE WARHEADS	-								
WB1038	IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)	1*								
02	MINE COMPONENTS (MINIMUM BAND)	16N7L-01	1675	212	3332	3556	1818	2817	11817	
SM2032	AFV - MINE DISPERSING VEHICLE	2*								
E40601	GROUND IMP MINE SCAT SYS AP M74(MYP)	2*								
E40602	GROUND IMP MINE SCAT SYS AT M75(MYP)	2*								
E71200	MODULAR PACK MINE SYSTEM	2*								
E75300	WIDE AREA SIDE PENETRATING MINE (WASPM)	2*								
Q39100	DISPENSER MINE XM139	2*								
WB1038	IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)	1*								
02	MINE COMPONENTS (AT RISK BAND)	16N7L-02	1275	711	2633	400				
SM2032	AFV - MINE DISPERSING VEHICLE	2*								
E40601	GROUND IMP MINE SCAT SYS AP M74(MYP)	2*								
E40602	GROUND IMP MINE SCAT SYS AT M75(MYP)	2*								
E71200	MODULAR PACK MINE SYSTEM	2*								
E75300	WIDE AREA SIDE PENETRATING MINE (WASPM)	2*								
Q39100	DISPENSER MINE XM139	2*								
WB1038	IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)	1*								
02	LANDMINE WARFARE	-								
SM2032	AFV - MINE DISPERSING VEHICLE	2*								
E40601	GROUND IMP MINE SCAT SYS AP M74(MYP)	2*								
E40602	GROUND IMP MINE SCAT SYS AT M75(MYP)	2*								
E71200	MODULAR PACK MINE SYSTEM	2*								
E75300	WIDE AREA SIDE PENETRATING MINE (WASPM)	2*								
Q39100	DISPENSER MINE XM139	2*								
WB1038	IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)	1*								
TOTAL FUNDS FOR PROJECT: FUNDED										
UNFUNDED										
			1675	212	4507	4267	4451	3217	11817	
			(1393)	(1733)	(1549)	(2793)	(1393)	(1733)	(1549)	(2793)

Legend
* - System designated in Mission Area

***** U N C L A S S I F I E D *****

Tue Jul 29 1986 11:57:58

projwkpblk

This report produces a project-workpackage table like the report projwkp,
but the funding numbers for the workpackages are omitted.

expr [first pg#] : acego ../neil/projwkpblk [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

PROJECT/WORKPACKAGE SUMMARY

COMMAND: ARDEC PROJECT: 63606 D006 TITLE: LANDMINE WARFARE DEVELOPMENT

TASK/WKP/SSN	WORKPACKAGE/SYSTEM TITLE	PDIP	FY87	FY88	FY89	FY90	FY91	FY92	FY93
01	03346 MINE SENSORS	-							
	E71200 MODULAR PACK MINE SYSTEM								
	E72193 CANISTER MINE (VOLCANO) XMB7								
	WB1038 IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)								
01	03350 MINE WARHEADS	-							
	WB1038 IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)								
02	03347 MINE COMPONENTS (MINIMUM BAND)	16M7L-01							
	5M2032 AFV - MINE DISPERSING VEHICLE								
	E40601 GROUND IMP MINE SCAT SYS AP M74(MYP)								
	E40602 GROUND IMP MINE SCAT SYS AT M75(MYP)								
	E71200 MODULAR PACK MINE SYSTEM								
	E75300 WIDE AREA SIDE PENETRATING MINE (WASPM)								
	Q39100 DISPENSER MINE XM139								
	WB1038 IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)								
02	13347 MINE COMPONENTS (AT RISK BAND)	16M7L-02							
	5M2032 AFV - MINE DISPERSING VEHICLE								
	E40601 GROUND IMP MINE SCAT SYS AP M74(MYP)								
	E40602 GROUND IMP MINE SCAT SYS AT M75(MYP)								
	E71200 MODULAR PACK MINE SYSTEM								
	E75300 WIDE AREA SIDE PENETRATING MINE (WASPM)								
	Q39100 DISPENSER MINE XM139								
	WB1038 IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)								
02	9334 LANDMINE WARFARE	-							
	5M2032 AFV - MINE DISPERSING VEHICLE								
	E40601 GROUND IMP MINE SCAT SYS AP M74(MYP)								
	E40602 GROUND IMP MINE SCAT SYS AT M75(MYP)								
	E71200 MODULAR PACK MINE SYSTEM								
	E75300 WIDE AREA SIDE PENETRATING MINE (WASPM)								
	Q39100 DISPENSER MINE XM139								
	WB1038 IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)								
TOTAL FUNDS FOR PROJECT:									
	FUNDED								
	UNFUNDED								

***** U N C L A S S I F I E D *****

Tue Jul 29 1986 11:58:35

sysdef

This report produces an unclassified listing of the systems in the control file and their BDP deficiencies and score statistics.

expr [first pg#] : acego ../qa/sysdef [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

CSS SYSTEM BDP RANKING AND DEFICIENCIES

SSN	DA MA	AMC MGR	TITLE	#DEF	#A	#B	#C	#D	#E	#X	BL	RATING
1. 1T0007	CSS	BELVOIR	JERRY CAN -REBUY/NDI	1	0	0	1	0	0	0	0	166.80
2. 1T0016	CSS	PM-PHL	PIP, EXTREME ENVIRON WATER SUPPLY	3	0	0	1	0	2	0	0	210.50
3. 1T0017	CSS	BELVOIR	PIP- WTR PURIFICATION UNIT REV OSMOSIS, 600 GPH	3	0	0	1	0	2	0	0	210.50
4. 1T0019	CSS	BELVOIR	PIP - SWA PETROLEUM DISTRI SYSTEM	1	0	0	1	0	0	0	0	166.80
5. 1T0123	CSS	BELVOIR	LIGHTWEIGHT, EXPANDABLE TOPHANDLER FOR 50K RT LT	2	0	0	0	2	0	0	0	162.80
6. 2A0001	CSS	AVSCOM	NON-DESTRUCTIVE TEST EQUIPMENT (NDTE)	6	1	2	2	1	0	0	0	1162.20
7. 2A0103	CSS	ARTA	HELICOPTER SNOW SKI SET (HSSS) (NEW FLEET)	2	0	0	0	0	2	0	0	81.60
J. 2M0004	CSS	AMCCOM	TEST STAND, ELECTRICAL VEHICULAR COMPONENT	1	0	1	0	0	0	0	0	124.00
9. 2M0012	CSS	TACOM	TEST STAND, HYDROVAC	1	0	1	0	0	0	0	0	124.00
10. 2M2001	CSS	TACOM	ARMORED MAINTENANCE VEHICLE (NEAR TERM)	12	1	3	3	5	0	0	0	2349.60
11. 2M2002	CSS	TACOM	ARMORED RESUPPLY MULTIPURPOSE SYSTEM	9	1	2	2	4	0	0	0	1901.00
12. 2M4003	CSS	TACOM	ALLIED KINETIC ENERGY RECOVERY ROPE (AKERR)	3	0	0	0	2	1	0	0	192.70
13. 2M4006	CSS	TACOM	TEST STAND, HEATER	2	0	1	0	0	1	0	0	163.30
14. 2T0001	CSS	BELVOIR	NDI WATERCRAFT	3	0	0	3	0	0	0	0	357.60
15. 3A0006	CSS	AVSCOM	UNIT MAINTENANCE AERIAL RECOVERY KIT (UMARK)	5	0	2	3	0	0	0	0	1017.20
16. 3A0007	CSS	AVSCOM	AIRCRAFT COMBAT MAINTENANCE/BATTLE DAMAGE REPAIR	7	1	2	1	3	0	0	0	1273.00
17. 3A0008	CSS	AVSCOM	ADV MATERIAL EXTERNAL CARGO SLING SYSTEM (AMCSS)	9	0	2	2	4	1	0	0	979.60
18. 3A0009	CSS	AVSCOM	HELICOPTER EXTERNAL CARGO SNUBBING SYSTEM (HEXCSS)	9	0	2	2	4	1	0	0	974.60
19. 3A0010	CSS	AVSCOM	HELICOPTER EXTERNAL AIR TRANSPORT (HEAT) KIT	8	0	2	2	3	1	0	0	924.40
20. 3A0011	CSS	AVSCOM	INTERNAL/EXTERNAL (INTEX) PALLET CARGO HANDLING SY	8	0	2	2	3	1	0	0	939.20
21. 3A0057	CSS	AVSCOM	A/C DECONTAMINATION, DEICING, AND CLEANING SYSTEM	5	1	0	4	0	0	0	0	1008.40
22. 3A0058	CSS	AVSCOM	NEW AIRCRAFT TOOL SYSTEM (NATS)	6	0	1	2	3	0	0	0	554.40
23. 3A0060	CSS	AVSCOM	TACTICAL A/C MAINTENANCE PLATFORM (TAMP)	4	0	0	2	2	0	0	0	331.40
24. 3A0061	CSS	AVSCOM	STANDARD AIRCRAFT TOWING TRACTOR SYSTEM	3	0	0	1	1	1	0	0	175.50

***** U N C L A S S I F I E D *****

Mon Jul 21 1986 09:20:27

sysdef2

This report produces an abbreviated version of sysdef, without the system titles or the score statistics. Only the ssn and deficiencies are printed.

expr [first pg#] ! acego ../qa/sysdef2 [pg letter] [code] [name] [command]

***** U N C L A S S I F I E D *****

EMW SYSTEM BDP DEFICIENCIES

SSN	DEFICIENCIES
1T0012	80- 133-B
3T0002	265- 315-B
3T0006	80- 133-B
3T0020	4-B 33-C
3T0023	95-D
3T0030	95-B
3T0046	41-E 307-E 313-D
3T0048	41-B 86-D 145-B 313-D 327-A
3T0049	29-C 41-B 86-D 145-B 313-D 327-B
3T0031	29-D 41-B 86-E 145-C 307-B 313-D 315-C 320-B
3T0033	29-E 41-C 145-C 229-D 307-E
3T0036	29-D 41-C 94-E 145-E 229-D 307-E 319-E
3T0037	29-B 41-B 86-C
3T0038	29-D 86-D 129-
3T0040	132-B 145-C
3T0062	1001-C
3T0063	0- 1001-C
3T0064	1001-C
3T0065	1001-B
3T0066	133-C
3T0067	132-C
3T0076	9-C 35-D 73-D 325-D
3T0079	4-C 95-B 282-B
3T0094	177-D 220-C 344-C
3T0097	4-D 37-D 110-D 170-E 175-
3T0112	6-D 8- 13- 45- 62-C 67-C 71-C 128-D 139-D 237-D 266-C 296-C
3T0127	35-C 110- 170-D 175-E 222-B 238-E 294-
3T0300	4-B 36-C 95-C 131-D 207-C 221-C 282-D 287-D
3T0301	95-C 116-C 131-C 221-D 234-C
3T0302	95-D 131-D
3T0303	35-E 95-E 131-E
3T0307	29-C 94-D
3T0003	29-C 41-A 86-B 106-A 133-C 149-A 229-A 313-A
3T0006	29-C 41-C 86-E 145-D 229-E 307-D
3T0007	29-C 41-C 145-C 229-D 307-D
3T0022	172-A 326-A
3T0027	80- 133-B
3T0038	17- 29-C 41-C 149-C 209- 213- 229-D 307-D 328-
3T0035	4- 85- 95-D 116- 131- 137- 241- 310-B
3T0036	80- 133-C
3T0037	29-B 41-D 86-C 94-E 105- 307-C
3T0038	29-C 41-C 86-D 129- 137-
3T0040	29-D 86-E 176-B 241-
3T0041	12- 29-E 86-D
6T0007	29-B 41-B 86-B 105- 129-
6T0013	4-C 110-C 170-D
6T0023	95-C 114- 120- 143-

***** U N C L A S S I F I E D *****

sysdollars

This report produces the confidential system rollup report for a single command and mission area with dollar values inserted.

```
expr [first pg#] ; acego ../neil/sysdollars [pg letter] [code] [name] [command] [ssn]
```

***** S A M P L E *****

SSN: WC5112

TITLE: COUNTEROBSTACLE VEHICLE (COV)

DA/TRADOC Mission Area: CS / EHM
 Commodity Line: CM
 Cross Functional Area: SUPPORT
 Acquisition Type: DEVELOPMENT

ANCMSC: TROSCOM
 ANC Manager: BELVOIR
 TRADOC Proponent: ENS
 Req. Document: (O&OP, DROC)

TOP 8 DEFICIENCIES: 29-C, 41-C, 86-B, 94-C, 132-D, 133-E, 306-C, 307-C

DESCRIPTION: REPLACEMENT FOR M728 COMBAT ENGINEERING VEHICLE (COV) IS A MULTIFUNCTIONAL ARMORED VEHICLE TO PROVIDE HEAVY FORCE WITH ENHANCED BATTLEFIELD MOBILITY. COV UTILIZES A CURRENT STATE-OF-THE-ART COMBINATION FULL VEHICLE WIDTH MINEPLOW/BULI DOZER BLADE, PADLOCKING SYSTEM, AND TWO MANEUVERABLE TELESCOPIC ARMS WITH VARIOUS QUICK CHANGE ATTACHMENTS. THE COV CAN RAPIDLY BREACH AND WORK MINEFIELDS MARKS USING THE MINEPLOW CONSTRUCT COMBAT TRAILS, TANK FIGHTING POSITION AND ANTI-TANK DITCHES WITH DOZER BLADE. URBAN RUBBLE AND OTHER OBSTACLES ARE OVERCOME USING THE COMBINATION PLOW/DOZER & TELESCOPIC ARM

PRODUCTION PROGRAM (\$K)

PDP/INCR	94	95	96	97	98	99	86	87	88	89	90	91	92	93
65BK-01														

DEVELOPMENT PROGRAMS (\$K)

	86	87	88	89	90	91	92	93
65810 DE63 E85 BELV 5011 MACI-PRESER OF ENGS, MIL EQUIP								
64COV DCOV 01 BELV 1316 NEW PROPOSED PROGRAM FOR COV								
63721 DJ30 40 TACO 400 COLLECTIVE PROT MATERIEL CONCEPTS/ARMORED VEHICLES		335	414	(7000) 431	(13000) 449	(13000) 467	(8000) 850	1500
63619 DL01 31 ARDE 01346 COUNTEROBSTACLE VEHICLE (COV)								
63619 DL01 31 ARDE 91346 COUNTEROBSTACLE VEHICLE (COV)								

TECH BASE PROGRAMS (\$K)

	86	87	88	89	90	91	92	93
63102 D071 01 LABC 3303 TRACK AND SUSPENSION		224	316	354				

Legend:

- v - Workpackage Contribution Value
- * - EHM Funded Workpackage
- m - Workpackage Supports Multiple Systems

***** S A M P L E *****

Thu Aug 7 1986 11:31:40

sysindex

This report produces the unclassified system index in alphabetical order by system title for a single mission area.

expr [first pg#] : acego ../neil/sysindex [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

EMW SYSTEMS IN ALPHABETICAL ORDER BY TITLE

SSN	DA	MA	TDC	MA/PROP	SYSTEM TITLE	AMC MGR	ACQ	PDIP	PAGE
005501	CS		EMW	ENS	0.5 - 10KW LOW NOISE GENERATOR SETS	AMCCOM	TC		
3T0019	CSS		EMW	ENS	1.5/3KW POWER CONDITIONING MODULE	PM-MEP	DEV	6S7G-01	
3T0030	CSS		EMW	EMS	10KW SILENT POWER PLANT	BELVOIR	DEV	6S7G-01	
3T0013	CSS		EMW	ENG	15/30/60 KW SIGNATURE SUPPRESSED GEN SETS	PM-MEP	DEV	6S7G-01	
3T0078	CSS		EMW	ENS	30 & 60KW POWER CONDITIONERS	PM-MEP	DEV	6S7G-01	
3T0025	CSS		EMW	ENS	3KW FREE PISTON STIRLING ENGINE GENERATOR SET	BELVOIR	DEV	6S7G-01	
3T0079	CSS		EMW	ENS	ADVANCED GENERATOR SET COMPONENTS	BELVOIR	DEV	6S7F-01	
3T0301	CSS		EMW	ENS	ADVANCED MINE CLEARING SYS	BELVOIR	DEV	6M7L-01	
3T0046	CS		EMW	ENS	ADVANCED MINE OBSTACLE SYSTEM (AMDS)	PM-MCD	DEV	6M7L-02	
600021	CS		EMW	ENS	ADVANCED PLASTIC EXPLOSIVE	PM-MCD	DEV	6R8H-01	
600010	CS		EMW	ENG	ADVANCED TACTICAL POWER SOURCES	PM-MCD	DEV	6R8H-01	
6E0001	CSS		EMW	ENS	AFV - COMBAT EARTHMOVER (CEM)	LABCOM	BBTA	6S7G-01	
5M2017	CS		EMW	ENS	AFV - MINE DISPERSING VEHICLE	TACOM	DEV	6M7Y-01	
5M2032	CS		EMW	ENS	AFV-COMBAT EXCAVATOR (CEV)	TACOM	DEV	6M7Y-01	
5M2016	CS		EMW	ENS	AFV-COMBAT MOBILITY VEHICLE (CMV)	TACOM	DEV	6M7Y-01	
5M2018	CS		EMW	ENS	AFV-ENGINEER SQUAD (SAPPER)	TACOM	DEV	6M7Y-01	
5M2015	CS		EMW	ENS	AFV-IMPROVED HEAVY ASSAULT BRIDGE (IHAB)	TACOM	DEV	6M7Y-01	
5M2014	CS		EMW	ENS	AIR CONDITIONERS VARIOUS SIZE/CAPACITIES	TACOM	DEV	6M7Y-01	
MF9300	CSS		EMW	ENS	AIRBORNE MINEFIELD DETECTION & RECONN SYS--AMIDARS	BELVOIR	DEV	6M7L-01	
5T0003	CB		EMW	ENS	ARMORED FAMILY OF VEHICLES (AFV)	TACOM	DEV	6M7Y-01	
5M2000	CC		CCH	ARM	ARMORED VEH LAUNCH BRIDGE (AVLB) - PIP	TACOM	PIP	FL61-01	
Q23000	CC		EMW	ENS	ARMY 21 GENERATOR SET	TACOM	PIP	FL61-01	
3T0309	CS		EMW	ENS	AUTOMATED COUNTERBARRIER SYSTEM	PM-MEP	DEV		
5T0037	CS		EMW	ENS	BARRIERS-BROAD BASE TECH AREA	BELVOIR	BBTA	6M7L-01	
6T0025	CS		EMW	ENS	BATTERY CHARGER PP-7286/U	BELVOIR	BBTA	6M7L-01	
881100	CSS		EMW	ORS	BRIDGE & ROAD MUNITION (BRM)	LABCOM	BC	6V3H-01	
E79501	CSS		EMW	ENS	BRIDGE AND ROAD MUNITION, IMPROVED (IBRM)	PM-MCD	DEV	6M7L-01	
6D0008	CS		EMW	ENS	BRIDGE ERECTION BOAT, SD	PM-MCD	DEV	6R8H-01	
M23601	CS		EMW	ENS	BRIDGE, BAILEY, M2	BELVOIR	TC	6XES-01	
M25001	CS		EMW	ENS	BRIDGE, ERECTION SET, MED GIRD	BELVOIR	TC	6XES-03	
M29700	CS		EMW	ENS	BRIDGE, FIXED, HIGHWAY, ALUM, MED GIRDER	BELVOIR	TC	6XES-01	
M29500	CS		EMW	ENS	BUILDING, PRE-FAB, RELOCATABLE	BELVOIR	TC	3R7J-01	
MA9160	CSS		EMW	ENS	CAMOUFLAGE SYSTEM	BELVOIR	TC	6R8H-01	
5T0035	CS		EMW	INF	CANETIP	BELVOIR	TC	6XES-01	
5T0028	CS		EMW	ENS	CANISTER MINE (VOLCANO) XMB7	BELVOIR	TC	6R8H-01	
E72195	CSS		EMW	ENS	CANISTER MINE PRACTICE XMBB (VOLCANO)	BELVOIR	TC	6R8H-01	
E72194	CSS		EMW	ENS	CAP, BLASTING, M6 SPC ELEC	AMCCOM	DEV	6S5C-01	
E52500	CSS		EMW	ENS	CAP, BLASTING SPEC NON-ELECT M7	AMCCOM	TC	6S5C-01	
E52600	CSS		EMW	ENS	CH DML BLK PENT 2LB M118	AMCCOM	TC	6V8B-01	
E50300	CSS		EMW	ENS	CHARGE DEMO SHAPED 40LB M3	AMCCOM	TC	6S5C-01	
E33200	CSS		EMW	ENS	CHARGE, DEMO, 40LB, CRATERING	AMCCOM	TC	6S5C-01	
E53100	CSS		EMW	ENS	CHG DEMO BLK TNT 1 LB	AMCCOM	TC	6S5C-01	
E51300	CSS		EMW	ENS	CLASS 60 FLOAT BRG	ARDEC	TC	6S5C-01	
E51000	CSS		EMW	ENS	COMBAT ENGINEER COMPONENTS	ARDEC	TC	6V8B-01	
M25301	CS		EMW	ENS		BELVOIR	TC		
5T0027	CS		EMW	ENS		BELVOIR	TD	6R8B-01	

***** U N C L A S S I F I E D *****

sysindex1

This report produces the unclassified system index in numerical order by SSN for a single mission area.

expr [first pg#] : acego ../neil/sysindex1 [pg letter] [code] [name]

***** U N C L A S S I F I E D *****

EMW SYSTEMS IN NUMERICAL ORDER BY SSN

SSN	DA	MA	TDC	MA/PROP	SYSTEM TITLE	AMC MGR	ACQ	PDIP	PAGE
1M1006	CC		EMW	ENS	COMBAT ENGINEER VEHICLE	TACOM	BC	FL61-01	
1M1101	C5		EMW	ENS	MINE PLOW BLADE	PM-MCD	PIP	6XLO-01	
1T0012	C5		EMW	ENS	RIBBON BRIDGE, IMPROVED	BELVOIR	PIP	6XE5-01	
3E0019	C5S		EMW	ENS	POWER CONVERTERS	LABCOM	DEV	6S70-01	
3M0201	C5		EMW	ENS	SEE VARIANT	TACOM	DEV	6S5E-01	
3T0002	C5		EMW	ENS	TUNNEL DETECTION SYSTEM (TUDES)	BELVOIR	DEV	6XE5-01	
3T0006	C5		EMW	ENS	HEAVY DRY SUPPORT BRIDGE	BELVOIR	DEV	6XE5-01	
3T0013	C5S		EMW	ENO	10KW SILENT POWER PLANT	PM-MEP	DEV	6S70-01	
3T0019	C5S		EMW	ENS	0.5 - 10KW LOW NOISE GENERATOR SETS	PM-MEP	DEV	6S70-01	
3T0020	C5S		EMW	ENS	INTEGRATED POWER ENVIRONMENTAL CONTROL SYSTEMS	BELVOIR	DEV	6S70-01	
3T0023	C5S		EMW	ENS	30 & 60KW POWER CONDITIONERS	BELVOIR	DEV	6S70-01	
3T0030	C5S		EMW	ENS	1.5/5KW POWER CONDITIONING MODULE	BELVOIR	DEV	6S70-01	
3T0046	C5		EMW	ENS	ADVANCED MINE CLEARING SYS	BELVOIR	DEV	6M7L-01	
3T0048	C5		EMW	ENS	LIGHT INFANTRY MINEFIELD BREACHING SYSTEM (LIMBS)	BETA	DEV	6M7L-01	
3T0049	CB		EMW	ENS	IMPROVED LINE CHARGE	BELVOIR	DEV	6M7L-01	
3T0051	C5		EMW	ENS	IMPROVED HAND HELD MINE DETECTOR	BELVOIR	DEV	6M7L-01	
3T0055	C5		EMW	ENS	TACTICAL OBSTACLE MARKING SYSTEM (TOMS)	BELVOIR	DEV	6M7L-01	
3T0056	C5		EMW	ENS	REACTIVE FRAGMENT MINE NEUTRALIZATION SYSTEM (FRAG)	BELVOIR	TD	6M7L-01	
3T0057	C5		EMW	ENS	WIDE-AREA NEUTRALIZATION MINE DEVICE (WAND)	BELVOIR	TD	6M7L-01	
3T0058	C5		EMW	ENS	MAGNETIC SIGNATURE SILENCING SYSTEM (MASSIS)	BELVOIR	DEV	6M7L-01	
3T0060	CC		EMW	ENS	OVERHEAD COVER (OHC) FOR INDIV FIGHTING POSITION	BELVOIR	TD	6XE5-01	
3T0062	C5		EMW	ENS	MACI COMML CONST EQUIP (CCE)	BELVOIR	DEV	6M72-01	
3T0063	C5		EMW	ENS	MACI HYDRAULIC SYS & COMP	BELVOIR	DEV	6M72-01	
3T0064	C5		EMW	ENS	CONSTRUCTION & COMBAT SUPPORT EQUIPMENT	BELVOIR	DEV	6M72-01	
3T0065	C5		EMW	ENS	MACI-HIGH SPEED EXCAVATOR	BELVOIR	DEV	6M72-01	
3T0066	C5		EMW	ENS	DUAL ROLE BRIDGE	BELVOIR	TD	6XE5-01	
3T0067	CC		EMW	ENS	FIELD FORTIFICATIONS	BELVOIR	TD	6XE5-01	
3T0076	C5S		EMW	ENS	DIVING & SUPPLY EQUIP FOR NBC	BETA	DEV	6X35-01	
3T0078	C5S		EMW	ENS	15/30/60 KW SIGNATURE SUPPRESSED GEN SETS	BELVOIR	DEV	CP6C-02	
3T0077	C5S		EMW	ENS	3KW FREE PISTON STIRLING ENGINE GENERATOR SET	PM-MEP	DEV	6S70-01	
3T0094	C5S		EMW	ENS	HEATER, DUCT TYPE, 250,000 BTUH	BELVOIR	DEV	6S70-01	
3T0097	C5S		EMW	ENS	INTEGRATED CHEMICAL FILTER ENVIRONMENTAL EQUIP-ICE	BELVOIR	DEV	6M8K-01	
3T0112	C5		EMW	ENS	FIELD ARMY MAPPING SYSTEMS (DTSS)	BELVOIR	DEV	6M70-01	
3T0127	C5S		EMW	ENS	FAMILY OF TOTAL ENVIRONMENTAL CONTROL SYS (TECS)	BELVOIR	DEV	6M8K-01	
3T0300	C5S		EMW	ENS	GEN INTEG SYS FOR TACTICAL VEHICLES	BELVOIR	DEV	6S70-01	
3T0301	C5S		EMW	ENS	ADVANCED GENERATOR SET COMPONENTS	BELVOIR	DEV	6S7F-01	
3T0302	C5S		EMW	ENS	ZINC-BROMIDE MODULE	BELVOIR	DEV	6S70-03	
3T0303	C5S		EMW	ENS	NBC PROTECTED POWER PLANTS	BELVOIR	DEV	-	
3T0304	C5		EMW	ENS	SURVIVABLE TACTICAL ARMY GENERATOR	PM-MEP	DEV	-	
3T0305	C5		EMW	ENS	ARMY 21 GENERATOR SET	PM-MEP	DEV	-	
3T0306	C5		EMW	ENS	INTEGRATED VEHICLE/GENERATOR	PM-MEP	DEV	-	
3T0307	C5		EMW	ENS	ELECTRIC EXPLOSIVE DETONATION SYSTEM	BETA	DEV	6M7L-01	
3M2000	CC		CCH	ARM	ARMORED FAMILY OF VEHICLES (AFV)	TACOM	DEV	6M7Y-01	
3M2014	C5		EMW	ENS	AFV-IMPROVED HEAVY ASSAULT BRIDGE (IHAB)	TACOM	DEV	6M7Y-01	
3M2015	C5		EMW	ENS	AFV-ENGINEER SQUAD (SAPPER)	TACOM	DEV	6M7Y-01	
3M2016	C5		EMW	ENS	AFV-COMBAT EXCAVATOR (CEV)	TACOM	DEV	6M7Y-01	

***** U N C L A S S I F I E D *****

B- 1

Fri Jul 11 1986 16:15:42

sysres

This report produces the unclassified summary of underresourced systems.
These are development systems without active 6.3B or 6.4 workpackages.

```
expr [first pg#] ; acego ../neil/sysres [pg letter] [code] [name] [command]
```

***** U N C L A S S I F I E D *****

SYSTEMS WITHOUT ACTIVE 6 38/6.4 WORKPACKAGES FOR CSS

DEVELOPMENT SCHEDULE

SSN	TITLE	DA/TDC	MA	FY86	FY87	FY88	FY89	FY90	FY91	FY92	FY93
1T0007	JERRY CAN --REBUY/NDI										
1T0123	LIGHTWEIGHT, EXPANDABLE TOPHANDLER FOR 50K RT										
2A0001	NON-DESTRUCTIVE TEST EQUIPMENT (NDTE)										
2A0103	HELICOPTER SNOW SKI SET (HSSS) (NEW FLEET)										
2N0012	TEST STAND, HYDROVAC										
2M4003	ALLIED KINETIC ENERGY RECOVERY ROPE (AKERR)										
2M4006	TEST STAND, HEATER										
3A0061	STANDARD AIRCRAFT TOWING TRACTOR SYSTEM										
3C0002	COMBAT SERVICE SUPPORT CONTROL SYSTEM (CSSC2)										
3C0003	TEST PROGRAM SET (TPS) STANDARDS & TOOLS										
3R0102	MEDICAL DEVELOPMENT SYSTEM										
3T0003	FIELD FUEL QUALITY MONITOR TEST										
3T0070	AIRDROP R.F.FUEL MOD DRUM										
3T0071	UNDERWATER FUEL STORAGE										
3T0092	PUMP, CAPTURED FUELS										
3T3374	IMPROVED NIGHT MAINTENANCE TECHNOLOGY										
4T5302	COMBAT VEHICLE CREWMAN'S ENSEMBLE										
5T0002	FIELD FUEL PRODUCTION UNIT										
6M5008	IMPROVED LIGHTWEIGHT LIFTING DEVICES										
4H0001	METROLOGY AND CALIBRATION										
9D0003	TEST STAND, TRANSMISSION										
9D0005	TEST STAND, ENGINE DYNAMOMETER										
8D3950	AMC INFORMATION PROCESSING EQUIPMENT (IPE)										
8T3270	TMDE MODERNIZATION										
D08900	FAMILY HEAVY TACTICAL VEHICLES- TRAILERS										
D40100	SNOW VEHICLE, TRACKED, CARGO/PERSONNEL										
M63900	PUMP ASSY LIQ GAS WHL 4 IN OUT 350 GPH										
M85900	MODULAR FIELD KITCHEN										
R32300	BOAT, SERVICE, CHI										
S60000	STEAM CLEANER										
WD501D	NUCLEAR SECURITY VISUAL ASSESSMENT EN H										

Legend: Funded (Tech Base Only)
 ----- Partial Funded
 ----- Unfunded

***** U N C L A S S I F I E D *****

Thu Jul 17 1986 16:08:25

unlkwkp

This unclassified report shows the workpackages which are not uniquely linked to a system and pdip for fiscal accountability.

expr [first pg#] | acego .. /neil/unlkwkp [pg letter] [code] [name] [command]

***** U N C L A S S I F I E D *****

CSS FUNDED WORKPACKAGES NOT LINKED TO SYSTEMS
WITH FISCAL RESOURCES SHOWN

DEVELOPMENT SCHEDULE

PE/PROJ/TASK/WPKO	TITLE	FY86	FY87	FY88	FY89	FY90	FY91	FY92	FY93
1.	23754 D420 01 4200104 ATE P3I PROGRAM SUPPORT						250	500	
2.	61102 A31B 80 31B8001 SECOND GEN UNCOOLED FOCAL PLANE			500	600	600			
3.	61102 A31B 80 31B8002 SPECIAL PURPOSE SENSOR	880	355						
4.	61102 A31B 80 31B8003 MULTI-SPECTRAL FOCAL PLANE		200	400	250	400	1500	1500	
5.	61102 A31B CO 31B8001 ELECTRO ACOUSTO-OPTIC MATERIALS				200	500	500		
6.	61102 A31B CO 31B8002 OPTICAL MAT'L RESEARCH			(275)	(814)	(890)			
7.	61102 A31B CO 31B8003 OPTICAL SIGNAL PROCESSING (UNFUNDED)						1000	1500	
8.	61102 A31B CO 31B8004 ADV OPTICS TECHNOLOGY		750	400					
9.	61102 A31B DO 31B8001 AUTONOMOUS SYSTEM UNDERSTANDING		860	480	300				
10.	61102 A31B DO 31B8002 SCENE ANALYSIS RESEARCH			200	750	500			
11.	61102 A31B DO 31B8003 FORMATION TARGETING			250	650	300			
12.	61102 A31B DO 31B8004 CLUTTER REJECTION					(729)	(2484)		
13.	61107 A31B DO 31B8005 AI SITUATION ANALYSTS (UNFUNDED)			800	738	300			
14.	61102 A31B HO 31B8001 EPI MATL TECH F/2ND GEN SYSTEMS			600	420				
15.	61102 A31B HO 31B8002 MATERIALS CHARACTERIZATION	265	229	600	420				
16.	61102 A31B HO 31B8003 DEVICE MODELLING	150	150	300	300				
17.	61102 A31B HO 31B8004 AUTO MATLS GROWTH TECH			200	200		1000		
18.	61102 A31B HO 31B8005 MONOLITHIC GROWTH			150	150	921	1030	500	
19.	61102 A31B HO 31B8006 SUPERLATTICE GROWTH						950	1748	
20.	61102 A31B MO 31B8001 WIDE BAND TUNABLE LAS RSCH (VIS/NIR/FIR)	1034	983	600	400				

***** U N C L A S S I F I E D *****

Mon Jul 28 1986 13:34:26

wkpindex

The unclassified workpackage index.

expr [first pg#] | acego ../neil/wkpindex [pg letter] [code] [name] [command]

***** U N C L A S S I F I E D *****

EMM ROTE PROJECT INDEX

PE	PROJ	TITLE	COMMAND	PAGE
63102	DL01	COMBAT ENG COMPONENTS	BELVOIR	----
63606	D006	LANDMINE HARFARE DEVELOPMENT	ARDEC	----
63606	D608	COUNTERMINE AND BARRIER DEVELOPMENT	BELVOIR	----
63619	D005	LANDMINE SYSTEMS	ARDEC	----
63619	D343	ROBOTIC OBSTACLE BREACHING ASSAULT TANK	TACOM	----
63619	D606	COUNTERMINES AND BARRIERS	ARDEC	----
63619	D607	EXPLOSIVE DEMOLITION SYSTEMS	ARDEC	----
63619	DL01	COUNTER OBSTACLE VEHICLE	ARDEC	----
63702	D010	TACTICAL POWER SOURCES	LABCOM	----
63702	D011	ADVANCED ELECTRICAL ENERGY CONCEPTS	BELVOIR	----
63712	D580	FIELD ARMY MAPPING	BELVOIR	----
63726	D001	DIGITAL TOPOGRAPHIC SUPPORT SYSTEM	BELVOIR	----
63726	DK39	GENERAL SPT EQUIPMENT	BELVOIR	----
63726	DK39	GENERAL SPT EQUIPMENT	PM-PML	----
64612	D021	EXPLOSIVE DEMOLITIONS	ARDEC	----
64612	D413	MINE DETECTION AND NEUTRALIZATION	ARDEC	----
64619	D016	MINE SYSTEMS	ARDEC	----
64714	D194	ENGINE DRIVEN GENERATORS	MEP	----
64714	D196	SILENT POWER GENERATING SOURCES	MEP	----
64716	D379	QUIK REACTION MULTICOLOR PRINTER	BELVOIR	----
64717	DH01	COMBAT ENGINEER EQUIPMENT	BELVOIR	----
64717	DL39	GENERAL SUPPORT EQUIP	BELVOIR	----
64717	DL39	GENERAL SUPPORT EQUIPMENT	PM-PML	----
64CDV	DC0V	COUNTEROBSTACLE VEHICLE - NEW PROPOSED 6.4 PROGRAM	BELVOIR	----

Tue Jul 29 1986 11:54:43

***** U N C L A S S I F I E D *****

wkpsys

This diagnostic report shows the linkages between workpackages and systems, sorted by workpackage, and shows which systems are in the control file, the pdip's that the systems are in, the system pdip and bdp priorities, and the unique ssn-pdip combination that the workpackage is linked to for fiscal accountability.

expr [first pg#] : acego ../neil/wkpsys [pg letter] [code] [name] [proj]

***** S A M P L E *****

PROJECT/WORKPACKAGE/SYSTEM LINKAGE SUMMARY

COMMAND	ARDEC	PROJECT: 63619	DO03	TITLE: LANDMINE SYSTEMS	PRI	BDP
02	01312	ICOMS AD EFFORT (MIN)		V^^	6M7L-01	BDP
	W1038	IMPROVED CONVENTIONAL MINE SYSTEM (ICOMS)		1*^	6M7L-01	495.20
04	01303	VOLCANO AD EFFORT (MIN)		V^^	6M7L-01	BDP
	5M2032	AFV - MINE DISPENSING VEHICLE		2**	6M7Y-01	806.60
	E72195	CANISTER MINE (VOLCANO) XMB7		1*	653C-01	399.60
					6571-01	399.60
					6V8B-01	399.60
					6V8Q-01	399.60
					6VBJ-01	399.60
					6VBJ-03	399.60
					6VBK-01	399.60
					6VBK-03	399.60
					6V8G-01	399.60
					6V8G-03	399.60
					6V8S-01	399.60
					6V8S-03	399.60
04	91303	VOLCANO AD EFFORT		V^^	-	BDP
05	01315	WIDE AREA MINE (WAM) AD EFFORT (MIN)		V^^	6M7L-01	BDP
	5M2032	AFV - MINE DISPENSING VEHICLE		3*	6M7Y-01	806.60
	6D0020	WIDE AREA MINE (WAM)		1**	6M7L-02	388.40
05	11315	WIDE AREA MINE (WAM) AD EFFORT (RISK)		V^^	6M7L-02	BDP
	6D0020	WIDE AREA MINE (WAM)		1**	6M7L-02	388.40
06	01318	DUAL SENSOR MINE SYSTEM AD EFFORT		V^^	-	BDP
CC	01374	ADVANCED MINE OBSTACLE SYSTEM AD (MIN)		V^^	6M7L-01	BDP
DD	01372	SELECTABLE LIGHTWEIGHT ATTACK MUNITION (SLAM) AD (MIN)		V^^	6M7L-01	BDP
	6D0016	SELECTABLE LIGHTWEIGHT ATTACK MUNITION (SLAM)		1**	6RBP-01	589.80
DD	91375	SLAM - UNFUNDED		V^^	-	BDP
	6D0016	SELECTABLE LIGHTWEIGHT ATTACK MUNITION (SLAM)		2**	6RBP-01	589.80
XX	01319	CONTROLLABLE MINE SYSTEM AD EFFORT (MIN)		V^^	6M7L-01	BDP
	5M2032	AFV - MINE DISPENSING VEHICLE		3**	6M7Y-01	806.60
	6D0014	CONTROLLABLE MINE SYSTEM		1	6M7L-02	0.00
XX	11319	CONTROLLABLE MINE SYSTEM AD EFFORT (RISK)		V^^	6M7L-02	BDP
	6D0014	CONTROLLABLE MINE SYSTEM		1	6M7L-02	0.00

***** S A M P L E *****

Legend:
 s - System Designated in Mission Area
 ^ - Major Designated System for Fiscal Accountability
 v - Workpackage-System Rating Factor

Thu Aug 7 1986 11:59:16

wrkapdx

This report produces an unclassified or classified report depending on the workpackage description. It is the workpackage appendix.

expr [first pg#] : acego ../neil/wrkapdx [pg letter] [code] [name] [command]

***** U N C L A S S I F I E D *****

WORKPACKAGE SUMMARY DATA FOR BELVOIR / CED

PE/Project/Task: 63102 DJ01 01 : WP 1047
Funded by: ENH Transition Date: Title: COMBAT ENGINEER COMPONENTS

	FY86	FY87	FY88	FY89	FY90	FY91	FY92	FY93	Description
Funded:	776	0	0	938	4113	4651	1717	1717	DESIGN, FABRICATE AND TEST FULL SCALE COMPONENTS OF A MODULAR, HORIZONTALLY LAUNCHED DRY GAP BRIDGE WITH THIS TEST BED, DEVELOP COMPOSITE MATERIAL STRUCTURALLY INTER-CHANGEABLE ELEMENTS THAT OFFER QUANTUM IMPROVEMENTS OVER CURRENTLY AVAILABLE METALLIC STRUCTURAL COMPONENTS. DEVELOP MATERIAL INGREDIENTS FOR COMPOSITE CONSTRUCTION OF COMPONENTS WHICH EFFECTIVELY UTILIZE LOW COST TEXTILE FABRICATION TECHNIQUES.
Unfunded:	0	0	0	0	0	0	0	0	
Flags									
LRDAP	0	0	0	938	4113	4651	1717	1717	

1987 Workplan

Systems Supported (SSN/TITLE/DA MA)	CS
3T0066 DUAL ROLE BRIDGE	CS
5K2014 AFV-IMPROVED HEAVY ASSAULT BRIDGE (IHAB)	CS
3T0027 COMBAT ENGINEER COMPONENTS	CS
3T0036 HIGH MOBILITY BRIDGING	CS
MC102C HEAVY ASSLT BRIDGE (HAB)	CS

C 1 97

***** U N C L A S S I F I E D *****

Tue Jul 29 1986 11:38:16

APPENDIX D

SOURCE CODE FOR
CURRENTLY AVAILABLE MAMP REPORTS

{ Streamlined Base Case and Type Classified Systems }
{ This Report Produces a Chart of Funding. }

{ 2/5/86 to add the mission area and date-time stamp.
2/7/86 to include ssndesc modifications.
5/20/86 to add group parameter, and changed in order to be run in mamp.
5/29/86 to exclude base case and only show type classified.
7/24/86 removed group parameter, changed to unclass, added all pdips. }

database mamp end

define

```
variable total_fund type long
variable eval       type integer
variable pno        type integer
variable p1         type long
variable p2         type long
variable p3         type long
variable p4         type long
variable p5         type long
variable p6         type long
variable p7         type long
variable p8         type long
variable p9         type long
variable p10        type long
variable p11        type long
variable p12        type long
variable p13        type long
variable p14        type long
variable p15        type long
variable p16        type long
variable u1         type long
variable u2         type long
variable u3         type long
variable u4         type long
variable u5         type long
variable u6         type long
variable u7         type long
variable u8         type long
variable u9         type long
variable u10        type long
variable u11        type long
variable u12        type long
variable u13        type long
variable u14        type long
variable u15        type long
variable u16        type long
param[1] pletter   type character length 1
param[2] code      type integer
param[3] name      type character length 12
param[4] cmd       type character length 12
```

end

input

```
prompt for pno using "Please enter the starting page number > "  
end
```

```

output
  left margin 0
  right margin 132
  report to "basetc.out"
end

read into c
  scl_ssn
  where scl_code = code and scl_name = name
end

read into a
  c
  ssn_amc_mgr ssn_da_ma ssn_tradoc_ma
  where ssn_acq_code = 2 and ssn_amc_mgr matches cmd
  joining c.scl_ssn = ssn_ssn
end

read into b
  a
  ssndroll
  lrrdp_title lrrdp_pdip
  lrrdp_proc1 lrrdp_proc2 lrrdp_proc3 lrrdp_proc4 lrrdp_proc5
  lrrdp_proc6 lrrdp_proc7 lrrdp_proc8 lrrdp_proc9 lrrdp_proc10
  lrrdp_proc11 lrrdp_proc12 lrrdp_proc13 lrrdp_proc14 lrrdp_proc15
  lrrdp_proc16
  lrrdpri_pri
  joining a.scl_ssn = optional lrrdp_ssn
  and a.scl_ssn = optional ssnd_ssn
  and lrrdp_pdip = optional lrrdpri_pdip
end

sort by scl_ssn lrrdp_pdip end

format

page header
  let eval = 0
  print column 48, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 39, "STREAMLINED SUMMARY OF TYPE CLASSIFIED SYSTEMS FOR ", name
  skip 3 lines
  print column 90, "PROCUREMENT SCHEDULE"
  print "-----"
  print " : SSN", 10 spaces, "TITLE/DEFICIENCIES", column 57, "DA/TDC MA",
  column 67,
  " : 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00 01 !"
  print "-----"
  print "!", column 67, "!", column 132, "!"

on last record
  print "!", column 67, "!", column 132, "!"
  print "-----"

```

before group of scl_ssn

```
let p1 = 0
let p2 = 0
let p3 = 0
let p4 = 0
let p5 = 0
let p6 = 0
let p7 = 0
let p8 = 0
let p9 = 0
let p10 = 0
let p11 = 0
let p12 = 0
let p13 = 0
let p14 = 0
let p15 = 0
let p16 = 0
let u1 = 0
let u2 = 0
let u3 = 0
let u4 = 0
let u5 = 0
let u6 = 0
let u7 = 0
let u8 = 0
let u9 = 0
let u10 = 0
let u11 = 0
let u12 = 0
let u13 = 0
let u14 = 0
let u15 = 0
let u16 = 0
```

before group of lrrdp_pdp

```
if lrrdppri_pri < 1.0 or lrrdppri_pri > 800.0 then begin
  let u1 = u1+lrrdp_procf1
  let u2 = u2+lrrdp_procf2
  let u3 = u3+lrrdp_procf3
  let u4 = u4+lrrdp_procf4
  let u5 = u5+lrrdp_procf5
  let u6 = u6+lrrdp_procf6
  let u7 = u7+lrrdp_procf7
  let u8 = u8+lrrdp_procf8
  let u9 = u9+lrrdp_procf9
  let u10 = u10+lrrdp_procf10
  let u11 = u11+lrrdp_procf11
  let u12 = u12+lrrdp_procf12
  let u13 = u13+lrrdp_procf13
  let u14 = u14+lrrdp_procf14
  let u15 = u15+lrrdp_procf15
  let u16 = u16+lrrdp_procf16
end
else begin
```

```

let p1 = p1+lrrdp_proc1
let p2 = p2+lrrdp_proc2
let p3 = p3+lrrdp_proc3
let p4 = p4+lrrdp_proc4
let p5 = p5+lrrdp_proc5
let p6 = p6+lrrdp_proc6
let p7 = p7+lrrdp_proc7
let p8 = p8+lrrdp_proc8
let p9 = p9+lrrdp_proc9
let p10 = p10+lrrdp_proc10
let p11 = p11+lrrdp_proc11
let p12 = p12+lrrdp_proc12
let p13 = p13+lrrdp_proc13
let p14 = p14+lrrdp_proc14
let p15 = p15+lrrdp_proc15
let p16 = p16+lrrdp_proc16
end

```

```

after group of scl_ssn
let total_fund = p1+u1+p2+u2+p3+u3+p4+u4+p5+u5+p6+u6+p7+u7+p8+u8+
p9+u9+p10+u10+p11+u11+p12+u12+p13+u13+p14+u14+p15+u15+p16+u16
print " ", scl_ssn, 2 spaces, lrrdp_title[1,45], 1 space,
      ssn_da_ma, " / ", ssn_tradoc_ma, column 67, "!";
if scl_ssn matches "4*" then print column 93, "<STOCK FUNDED>", column 132, "!";
else if total_fund = 0 then print column 93, "<NOT SCHEDULED>", column 132, "!";
else begin
  if p1>0 then begin if u1>0 then print "*--"; else print "****"; end
  else begin if u1>0 then print "----"; else print " "; end
  if p2>0 then begin if u2>0 then print "*--"; else print "****"; end
  else begin if u2>0 then print "----"; else print " "; end
  if p3>0 then begin if u3>0 then print "*--"; else print "****"; end
  else begin if u3>0 then print "----"; else print " "; end
  if p4>0 then begin if u4>0 then print "*--"; else print "****"; end
  else begin if u4>0 then print "----"; else print " "; end
  if p5>0 then begin if u5>0 then print "*--"; else print "****"; end
  else begin if u5>0 then print "----"; else print " "; end
  if p6>0 then begin if u6>0 then print "*--"; else print "****"; end
  else begin if u6>0 then print "----"; else print " "; end
  if p7>0 then begin if u7>0 then print "*--"; else print "****"; end
  else begin if u7>0 then print "----"; else print " "; end
  if p8>0 then begin if u8>0 then print "*--"; else print "****"; end
  else begin if u8>0 then print "----"; else print " "; end
  if p9>0 then begin if u9>0 then print "*--"; else print "****"; end
  else begin if u9>0 then print "----"; else print " "; end
  if p10>0 then begin if u10>0 then print "*--"; else print "****"; end
  else begin if u10>0 then print "----"; else print " "; end
  if p11>0 then begin if u11>0 then print "*--"; else print "****"; end
  else begin if u11>0 then print "----"; else print " "; end
  if p12>0 then begin if u12>0 then print "*--"; else print "****"; end
  else begin if u12>0 then print "----"; else print " "; end
  if p13>0 then begin if u13>0 then print "*--"; else print "****"; end
  else begin if u13>0 then print "----"; else print " "; end
  if p14>0 then begin if u14>0 then print "*--"; else print "****"; end
  else begin if u14>0 then print "----"; else print " "; end
  if p15>0 then begin if u15>0 then print "*--"; else print "****"; end
  else begin if u15>0 then print "----"; else print " "; end

```



```

    if p16>0 then begin if u16>0 then print "*--"; else print "****"; end
    else begin if u16>0 then print "----"; else print " "; end
    print "!";
end
print " ";
if ssnd_def1>0 then begin print ssnd_def1 using " ####";
  if ssnd_con1<>" " then print "-",ssnd_con1; else print " "; end
if ssnd_def2>0 then begin print ssnd_def2 using " ####";
  if ssnd_con2<>" " then print "-",ssnd_con2; else print " "; end
if ssnd_def3>0 then begin print ssnd_def3 using " ####";
  if ssnd_con3<>" " then print "-",ssnd_con3; else print " "; end
if ssnd_def4>0 then begin print ssnd_def4 using " ####";
  if ssnd_con4<>" " then print "-",ssnd_con4; else print " "; end
if ssnd_def5>0 then begin print ssnd_def5 using " ####";
  if ssnd_con5<>" " then print "-",ssnd_con5; else print " "; end
if ssnd_def6>0 then begin print ssnd_def6 using " ####";
  if ssnd_con6<>" " then print "-",ssnd_con6; else print " "; end
if ssnd_def7>0 then begin print ssnd_def7 using " ####";
  if ssnd_con7<>" " then print "-",ssnd_con7; else print " "; end
if ssnd_def8>0 then begin print ssnd_def8 using " ####";
  if ssnd_con8<>" " then print "-",ssnd_con8; else print " "; end
print column 67,"!";column 132, "!"
let eval = eval+1
if eval = 20 then begin
  print "!",column 67,"!";column 132, "!"
  print "-----"
  skip to top of page
end

page trailer
print "Legend:",column 48,"***** U N C L A S S I F I E D *****"
print " **** - Funded Procurement",column 100,date,2 spaces,time
print " ---- - Unfunded Procurement";
if pletter = "Z" then print column 60,pno
else print column 60,pletter,"-",pno using "###"
let pno = pno+1

end

```

< 7/10/86. LRRDP/BDP alignment report for 1986 BDP deficiencies.
7/15/86. changed uniwkp_wc to uniwkp_srf. >

database mamp end

```
define
  variable cont      type integer
  variable pno       type integer
  variable flag      type integer
  param[1] pletter   type character length 1
  param[2] code      type integer
  param[3] name      type character length 12
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  left margin 0
  right margin 132
  report to "bdpalignmt.out"
end

read into d
  dcl_def
  defx_85
  where dcl_name = name and dcl_code = code
  joining dcl_def = defx_86
end

read into e
  d
  ssndef_ssn ssndef_cont_value
  where ssndef_cont_value <> " " and ssndef_cont_value <> "X"
  joining d.dcl_def = ssndef_def
end

read into c
  e
  lrrdp_sys_idx lrrdp_title
  ssn_tradoc_pro
  joining e.ssn_ssn = lrrdp_ssn
  and e.ssn_ssn = optional ssn_ssn
end

read into b
  c
  uniwkp_srf
  wkpkg_cmd wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no wkpkg_title
  joining c.lrrdp_sys_idx = optional uniwkp_sys_idx
  and uniwkp_wkpkg_idx = optional wkpkg_wkpsys_idx
end

sort by dcl_def lrrdp_pdip ssndef_ssn
  wkpkg_cmd wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no
```

```
end

format

page header
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 49, "LRRDP/BDP 1986 ALIGNMENT FOR ", name;
  if cont = 1 then print " (continued)" else print ""
  let cont = 1
  skip 2 lines
  print 2 spaces, "BDP DEFICIENCY: 1986:", dcl_def, 5 spaces, "1985:", defx_85
  skip 2 lines
  print 2 spaces, "PDIP-INCR", 10 spaces, "NSI/SSN/PE PROJ", 25 spaces,
    "PROGRAM", 25 spaces, "PROPONENT", 15 spaces, "CONT VALUE"
  print 1 space, "-----", 8 spaces, "-----", 23 spaces,
    "-----", 23 spaces, "-----", 13 spaces, "-----"

page trailer
  skip 2 lines
  print column 47, "***** U N C L A S S I F I E D *****"
  print column 100, date, 2 spaces, time
  if pletter="Z" then print column 60, pno
  else print column 60, pletter, "-", pno
  using "###" let pno = pno+1

before group of dcl_def
  let cont = 0
  skip to top of page

before group of lrrdp_pdip
  let flag = 0

before group of ssndef_ssn
  skip 1 line
  if flag = 0 then begin
    print 3 spaces, lrrdp_pdip[1, 4], "-", lrrdp_pdip[5, 6];
    let flag = 1
  end
  else print 10 spaces;
  print 11 spaces, ssndef_ssn, 15 spaces, lrrdp_title[1, 45], 9 spaces,
    ssn_tradoc_pro, 22 spaces, ssndef_cont_value

before group of wkpkg_no
  if wkpkg_cmd <> " " then begin
    print column 28, wkpkg_cmd[1, 8], 1 space, wkpkg_pe, 1 space, wkpkg_proj, 1 space,
      wkpkg_task[1, 4], 1 space, 1 space, wkpkg_no, 3 spaces, wkpkg_title[1, 50],
      8 spaces, uniwp_srf using "#"
  end

end
```

May 16 12 41 1986 bpcss Page 1

(Bill Payer Report

5/15/86 bill payer report taken from fundwppbdp2. Added running subtotals.
5/16/86 changed titles, added command parameter, changed the bpcss file.)

database camp end

define

```
variable pno      type integer
variable pgen     type integer
variable cnt      type integer
variable s0       type long
variable s1       type long
variable s2       type long
variable s3       type long
variable s4       type long
variable s5       type long
variable s6       type long
variable s7       type long
param[1] plat    type character length 1
param[2] cmd      type character length 12
```

end

input

```
prompt for pno using "Please enter the starting page number > "
end
```

output

```
left margin 0
right margin 132
report to "bpcss out"
end
```

read into b

```
bpcss
where bpcss_cmd matches cmd
end
```

sort by bpcss_bdp descending end

format

page header

```
print column 47, "***** U N C L A S S I F I E D *****"
skip 2 lines
print column 48, "CSS N TO 1 BDP WORKPACKAGE BILLPAYERS"
print column 48, "          WITH CUMULATIVE TOTALS"
skip 2 lines
print column 87, "FUNDED DEVELOPMENT SCHEDULE"
print "-----"
print "          PE/PROJ/TASK/WKPKG", 5 spaces, "TITLE", column 67,
print "  FY86   FY87   FY88   FY89   FY90   FY91   FY92   FY93  !"
print "-----"
print "-----"
print "!", column 67, "!", column 132, "!"
```

May 16 12:41 1986 bpcss Page 2

```

let pgend = 0

page trailer
if pgend = 0 then begin
  print "-----"
end
else skip 1 line
skip 2 lines
print column 47, "***** U N C L A S S I F I E D *****"
print column 100, date, 2 spaces, time
if pietter = "Z" then print column 60, pno
else print column 60, pietter, "-", pno using "####"
let pno = pno+1

on last record
print "-----"
let pgend = 1

before group of bpcss_bdp
print "!" bpcss_bdp using "### ", bpcss_title[1, 55], column 67, "!";
if bpcss_f0:0 then print bpcss_f0 using "##### "; else print " ";
if bpcss_f1:0 then print bpcss_f1 using "##### "; else print " ";
if bpcss_f2:0 then print bpcss_f2 using "##### "; else print " ";
if bpcss_f3:0 then print bpcss_f3 using "##### "; else print " ";
if bpcss_f4:0 then print bpcss_f4 using "##### "; else print " ";
if bpcss_f5:0 then print bpcss_f5 using "##### "; else print " ";
if bpcss_f6:0 then print bpcss_f6 using "##### "; else print " ";
if bpcss_f7:0 then print bpcss_f7 using "##### "; else print " ";
print "!"
print "!" 5 spaces, bpcss_cmd, 1 space,
      bpcss_pa, 1 space, bpcss_proj, 1 space, bpcss_task clipped, 1 space,
      bpcss_wkpkg, column 55, "CUM TOTAL:", column 67, "!";
let s0 = s0+bpcss_f0
let s1 = s1+bpcss_f1
let s2 = s2+bpcss_f2
let s3 = s3+bpcss_f3
let s4 = s4+bpcss_f4
let s5 = s5+bpcss_f5
let s6 = s6+bpcss_f6
let s7 = s7+bpcss_f7
print s0 using "##### ", s1 using "##### ", s2 using "##### ",
      s3 using "##### ", s4 using "##### ", s5 using "##### ",
      s6 using "##### ", s7 using "##### ", "!";
print "!" column 67, "!", column 132, "!"
end

```

(Reverse of Bill Payer Report - Decision Aid

5/15/86 bill payer report taken from fundwpbdp2. Added running subtotals.
5/15/86 changed titles, added command parameter, changed the bpcss file.
5/15/86 taken from bpcss bill payer report and changed title and sort)

database mamp end

define

variable pno type integer
variable ppend type integer
variable cnt type integer
variable s0 type long
variable s1 type long
variable s2 type long
variable s3 type long
variable s4 type long
variable s5 type long
variable s6 type long
variable s7 type long
param[1] pletter type character length 1
param[2] cmd type character length 12

end

input

prompt for pno using "Please enter the starting page number > "
end

output

left margin 0
right margin 132
report to "bocssa.out"
end

read into b

bpcss
where bpcss_cmd matches cmd
end

sort by bpcss_bdp end

format

page header

print column 47, "***** U N C L A S S I F I E D *****"
skip 2 lines
print column 47, "CSS 1 TO N BDP WORKPACKAGE DECISION AID"
print column 47, " WITH CUMULATIVE TOTALS"

skip 2 lines
print column 87, "FUNDED DEVELOPMENT SCHEDULE"

print "-----"
print " | PE/PROJ/TASK/WKPKG", 5 spaces, "TITLE", column 67, |"
print " | FY86 FY87 FY88 FY89 FY90 FY91 FY92 FY93 |"
print "-----"

```
print "!", column 67, "!", column 132, "!"
let pgend = 0

page trailer
if pgend = 0 then begin
  print "-----",
  end
else skip 1 line
skip 2 lines
print column 47, "***** U N C L A S S I F I E D *****"
print column 100, date, 2 spaces, time
if pletter = "Z" then print column 60, pno
else print column 60, pletter, "-", pno using "###"
let pno = pno+1

on last record
print "-----",
let pgend = 1

before group of bpcss_bdp
print "!", bpcss_bdp using "### ", bpcss_title[1,55], column 67, "!"
if bpcss_f0:0 then print bpcss_f0 using "##### "; else print " ";
if bpcss_f1:0 then print bpcss_f1 using "##### "; else print " ";
if bpcss_f2:0 then print bpcss_f2 using "##### "; else print " ";
if bpcss_f3:0 then print bpcss_f3 using "##### "; else print " ";
if bpcss_f4:0 then print bpcss_f4 using "##### "; else print " ";
if bpcss_f5:0 then print bpcss_f5 using "##### "; else print " ";
if bpcss_f6:0 then print bpcss_f6 using "##### "; else print " ";
if bpcss_f7:0 then print bpcss_f7 using "##### "; else print " ";
print "!"
print "!", 5 spaces, bpcss_cmd, 1 space,
      bpcss_pe, 1 space, bpcss_proj, 1 space, bpcss_task clipped, 1 space,
      bpcss_wkpkg, column 55, "CUM TOTAL:", column 67, "!"
let s0 = s0+bpcss_f0
let s1 = s1+bpcss_f1
let s2 = s2+bpcss_f2
let s3 = s3+bpcss_f3
let s4 = s4+bpcss_f4
let s5 = s5+bpcss_f5
let s6 = s6+bpcss_f6
let s7 = s7+bpcss_f7
print s0 using "##### ", s1 using "##### ", s2 using "##### ",
      s3 using "##### ", s4 using "##### ", s5 using "##### ",
      s6 using "##### ", s7 using "##### ", "!"
print "!", column 67, "!", column 132, "!"

end
```

(This program creates the bill payer record for a particular mission area.)

database mamp end

define

```
param[1] base type integer
param[2] code type integer
param[3] name type character length 12
variable cnt type integer
variable v type character length 1
variable byr type integer
variable f0 type long
variable f1 type long
variable f2 type long
variable f3 type long
variable f4 type long
variable f5 type long
variable f6 type long
variable f7 type long
```

end

output

```
top margin 0
page length 32000
left margin 0
right margin 132
report to "bpdata out"
```

end

read into a

```
pcl_idx
where pcl_code = code and pcl_name = name
and (pcl_subcat = "6.3A" or pcl_subcat = "6.3B" or pcl_subcat = "6.4"
or pcl_subcat = "6.7")
```

end

read into b

```
prior2_score prior2_nsys
wkpkg_no_idx wkpkg_pe wkpkg_proj wkpkg_task wkpkg_title
joining a pcl_idx = wkpkg_proj_idx
and wkpkg_no_idx = optional prior2_wkpkg_idx
```

end

read into d

```
wkpf_wkpkg_idx wkpf_yr wkpf_fund
where wkpf_fund > 0
joining b wkpkg_no_idx = wkpf_wkpkg_idx
```

end

read into a

```
b
d.wkpf_yr d.wkpf_fund
joining b wkpkg_no_idx = optional d.wkpf_wkpkg_idx
```

end

sort by prior2_score descending prior2_nsys descending


```
    wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no wkpf_yr end
```

```
format
```

```
before group of wkpkg_no
```

```
  let v = ""
```

```
  let byr = wkpf_yr
```

```
  let f0 = 0
```

```
  let f1 = 0
```

```
  let f2 = 0
```

```
  let f3 = 0
```

```
  let f4 = 0
```

```
  let f5 = 0
```

```
  let f6 = 0
```

```
  let f7 = 0
```

```
  let cnt = cnt+1
```

```
  print cnt, v, wkpkg_title, v, wkpkg_cmd, v, wkpkg_pe, v, wkpkg_proj, v,  
        wkpkg_task, v, wkpkg_no, v;
```

```
after group of wkpkg_no
```

```
  print f0, v, f1, v, f2, v, f3, v, f4, v, f5, v, f6, v, f7, v
```

```
before group of wkpf_yr
```

```
  if wkpf_yr = base then let f0 = wkpf_fund
```

```
  else if wkpf_yr = base+1 then let f1 = wkpf_fund
```

```
  else if wkpf_yr = base+2 then let f2 = wkpf_fund
```

```
  else if wkpf_yr = base+3 then let f3 = wkpf_fund
```

```
  else if wkpf_yr = base+4 then let f4 = wkpf_fund
```

```
  else if wkpf_yr = base+5 then let f5 = wkpf_fund
```

```
  else if wkpf_yr = base+6 then let f6 = wkpf_fund
```

```
  else if wkpf_yr = base+7 then let f7 = wkpf_fund
```

```
end
```

Jun 4 13:24 1986 cksyscmd Page 1

(System Check for a Command and a mission area
Modified 5/23/86. changed to mamp database.)

database mamp end

```
define
  variable total_fund type integer
  param[1] code       type integer
  param[2] name       type character length 12
  param[3] cmd        type character length 12
end
```

```
output
  left margin 0
  right margin 132
  report to "cksyscmd.out"
end
```

```
read into b
  scl_ssn
  where scl_code = code and scl_name = name
end
```

```
read into c
  b
  ssn hipri_title
  where ssn_amc_mgr = cmd
  joining b.scl_ssn = optional ssn_ssn
         b.scl_ssn = hipri_ssn
end
```

```
read into a
  c
  lrrdp_ssn
  lrrdp_procf1 lrrdp_procf2 lrrdp_procf3
  lrrdp_procf4 lrrdp_procf5 lrrdp_procf6
  lrrdp_procf7 lrrdp_procf8 lrrdp_procf9
  lrrdp_procf10 lrrdp_procf11 lrrdp_procf12
  lrrdp_procf13 lrrdp_procf14 lrrdp_procf15
  lrrdp_procf16
  ssndroll
  asssr
  joining c.scl_ssn = optional lrrdp_ssn
         and c.scl_ssn = optional ssnd_ssn
         and c.scl_ssn = optional asssr_ssn
end
```

```
read into b
  a
  wkpkg_pe wkpkg_proj wkpkg_task wkpkg_title
  wkpkg_subcat wkpkg_cmd wkpkg_no wkpkg_pri wkpkg_lab
  wkpkg_trans_date
  joining a.scl_ssn = optional wkpsys_ssn
         and wkpsys_wkpkg_idx = optional wkpkg_wkpsys_idx
end
```

```
sort by ssn_amc_mgr, scl_ssn, wkpkg_cmd, wkpkg_subcat descending,  
      wkpkg_pe, wkpkg_proj, wkpkg_no  
end
```

format

page header

```
print column 47, "***** C O N F I D E N T I A L *****"  
skip 2 lines
```

page trailer

```
skip 1 line  
print column 47, "***** C O N F I D E N T I A L *****"  
skip 1 line  
print 60 spaces, pageno
```

before group of scl_ssn

```
skip to top of page  
print 20 spaces, "SSN: ", scl_ssn, 20 spaces, "TITLE: ";  
if hipri_title = " " then print "_____"  
else print hipri_title
```

```
skip 1 line  
print "Associated SSN's:";  
if asr_ssn <> " " then print 4 spaces, asr_ssn1, 4 spaces, asr_ssn2,  
      4 spaces, asr_ssn3, 4 spaces, asr_ssn4, 4 spaces, asr_ssn5,  
      4 spaces, asr_ssn6, 4 spaces, asr_ssn7, 4 spaces, asr_ssn8,  
      4 spaces, asr_ssn9, 4 spaces, asr_ssn10  
else print 4 spaces, "_____", 4 spaces, "_____", 4 spaces, "_____"
```

```
skip 1 line  
print "Mission Area: ";  
if ssn_da_ma = " " then print "_____"  
else print ssn_da_ma;  
if ssn_tradoc_ma = " " then print "/_____"  
else print "/", ssn_tradoc_ma;
```

```
print column 80, "AMCMSC: ";  
if ssn_amc_msc = " " then print "_____"  
else print ssn_amc_msc
```

```
print "Commodity Line: ";  
if ssn_com_line = " " then print "_____"  
else print ssn_com_line;
```

```
print column 80, "AMC Manager: ";  
if ssn_amc_mgr = " " then print "_____"  
else print ssn_amc_mgr
```

```
print "Cross Functional Area: ";  
if ssn_xfuncarea = " " then print "_____"  
else print ssn_xfuncarea;
```

```
print column 80, "TRADOC Proponent: ";  
if ssn_tradoc_pro = " " then print "_____"  
else print ssn_tradoc_pro
```

```
print "Acquisition Type: ";  
if ssn_acq_code = 1 then print "Base Case";  
else if ssn_acq_code = 2 then print "Type Classified";  
else if ssn_acq_code = 3 then print "Development";  
else if ssn_acq_code = 4 then print "PIPs";  
else if ssn_acq_code = 5 then print "Technology Demonstrators";  
else if ssn_acq_code = 6 then print "Broad Base Tech Area";
```

```

    else if ssn_acq_code = 7 then print "Requirement Above Corps";
    else print " ";
print column 80,"Req. Document: ", "("";
if ssn_req_doc1=" " then print " _____"; else print ssn_req_doc1," ";
if ssn_req_doc2=" " then print " _____"; else print ssn_req_doc2," ";
if ssn_req_doc3=" " then print " _____)" else print ssn_req_doc3,")"
skip 1 line
print "DEFICIENCIES: ";
if ssn_ssn<>" " then begin
  if ssnd_def1>0 then print ssnd_def1,"-",ssnd_con1;
  if ssnd_def2>0 then print " ",ssnd_def2,"-",ssnd_con2;
  if ssnd_def3>0 then print " ",ssnd_def3,"-",ssnd_con3;
  if ssnd_def4>0 then print " ",ssnd_def4,"-",ssnd_con4;
  if ssnd_def5>0 then print " ",ssnd_def5,"-",ssnd_con5;
  if ssnd_def6>0 then print " ",ssnd_def6,"-",ssnd_con6;
  if ssnd_def7>0 then print " ",ssnd_def7,"-",ssnd_con7;
  if ssnd_def8>0 then print " ",ssnd_def8,"-",ssnd_con8;
  print " " end
else print " _____"
skip 2 lines
print "DESCRIPTION: ";
if ssn_ssn<>" " then begin
  print ssn_desc[1,120] clipped
  print 12 spaces,ssn_desc[121,240] clipped
  print 12 spaces,ssn_desc[241,360] clipped
  print 12 spaces,ssn_desc[361,480] clipped
  print 12 spaces,ssn_desc[481,600] clipped end
else begin
  print " _____"
  print 12 spaces," _____"
  print 12 spaces," _____"
  print 12 spaces," _____"
  print 12 spaces," _____"
end
skip 2 lines
print "PROCUREMENT FUNDING DATA"
if lrrdp_ssn<>" " then begin
  skip 1 line
  print "YEAR ",5 spaces,"FUNDED"
  print "1986 ",lrrdp_procf1
  print "1987 ",lrrdp_procf2
  print "1988 ",lrrdp_procf3
  print "1989 ",lrrdp_procf4
  print "1990 ",lrrdp_procf5
  print "1991 ",lrrdp_procf6
  print "1992 ",lrrdp_procf7
  print "1993 ",lrrdp_procf8
  print "1994 ",lrrdp_procf9
  print "1995 ",lrrdp_procf10
  print "1996 ",lrrdp_procf11
  print "1997 ",lrrdp_procf12
  print "1998 ",lrrdp_procf13
  print "1999 ",lrrdp_procf14
  print "2000 ",lrrdp_procf15
  print "2001 ",lrrdp_procf16
end
```

```

else begin
  skip 1 line
  print "YEAR          FUNDED  "
  print "1986          _____"
  print "1987          _____"
  print "1988          _____"
  print "1989          _____"
  print "1990          _____"
  print "1991          _____"
  print "1992          _____"
  print "1993          _____"
  print "1994          _____"
  print "1995          _____"
  print "1996          _____"
  print "1997          _____"
  print "1998          _____"
  print "1999          _____"
  print "2000          _____"
  print "2001          _____"
end
skip 2 lines
print "WORKPACKAGE DATA"
skip 1 line
print "PE      PROJ  TASK  SUB      WPKPG  TRAN  COMMAND  LAB      PRI      TITLE"
skip 1 line

on every record {before group of wkpsys_wkpkg}
if wkpkg_pe<>" " then print wkpkg_pe,2 spaces;
else print "_____";
if wkpkg_proj<>" " then print wkpkg_proj,3 spaces;
else print "_____";
if wkpkg_task<>" " then print wkpkg_task[1,4],2 spaces;
else print "_____";
if wkpkg_subcat<>" " then print wkpkg_subcat,2 spaces;
else print "_____";
if wkpkg_no<>" " then print wkpkg_no,2 spaces;
else print "_____";
if wkpkg_trans_date<>" " then print wkpkg_trans_date,2 spaces;
else print "_____";
if wkpkg_cmd<>" " then print wkpkg_cmd[1,8],2 spaces;
else print "_____";
if wkpkg_lab<>" " then print wkpkg_lab[1,8],2 spaces;
else print "_____";
if wkpkg_pri<>0 then print wkpkg_pri using "###",5 spaces;
else print "_____";
if wkpkg_title<>" " then print wkpkg_title
else print "_____";

end

```

(Check for Workpackages not linked to systems.

5/29/86. changed to mamp database.
7/24/86 revised to make generic.)

database mamp end

```
define
  param[1] code   type integer
  param[2] name   type character length 12
end
```

```
output
  left margin 0
  right margin 132
  report to "ckwpkg.out"
end
```

```
read into c
  pcl_idx
  where pcl_code = code and pcl_name = name
end
```

```
read into a
  wkpkg_wkpsys_idx wkpkg_lab wkpkg_title
  joining c.pcl_idx = wkpkg_proj_idx
end
```

```
read into b
  a
  joining a.wkpkg_wkpsys_idx = wkpsys_wkpkg_idx
end
```

assign c = a minus b end

sort by wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no end

format

```
page header
  print column 45, name clipped, " FUNDED WORKPACKAGES NOT LINKED TO SYSTEMS"
  skip 2 lines
  print "PE      PROJ      TASK      SUB      WKPKG      COMMAND LAB      TITLE"
  skip 1 line
```

```
page trailer
  skip 1 line
  print column 60, pageno
```

```
before group of wkpkg_no
  print wkpkg_pe, 2 spaces, wkpkg_proj, 3 spaces, wkpkg_task, 2 spaces,
        wkpkg_subcat, 2 spaces, wkpkg_no, 2 spaces,
        wkpkg_cmd[1,8], 1 space, wkpkg_lab, 2 spaces, wkpkg_title
```

end

database mamp end

```
{ Last modified 2/6/86 to add mission area and date-time stamp.  
  2.7.86 to change ssndesc_commodity.  
  3/6/86 to add page letter option and page number input.  
  5/6/86 changed to mamp database.  
  6/13/86 deleted command sort.  
  7/24/86 merged w/ comindex1 to have parameter to indicate whether to print  
    column for profile. }
```

define

```
param[1] pletter type character length 1  
param[2] code type integer  
param[3] name type character length 12  
param[4] pro type character length 1  
variable pno type integer
```

end

input

```
prompt for pno using "Please enter the starting page number > "  
end
```

output

```
left margin 0  
right margin 132  
report to "comindex.out"  
end
```

read into b

```
scl_ssn  
where scl_name = name and scl_code = code  
end
```

read into a

```
ssn_com_line  
joining b.scl_ssn = ssn_ssn  
end
```

sort by ssn_com_line end

format

before group of ssn_com_line

```
if pro = "Y" then print 40 spaces; else print 50 spaces;  
print ssn_com_line, 17 spaces, "----";  
if pro = "Y" then print 17 spaces, "----" else print ""
```

page header

```
print column 47, "***** UNCLASSIFIED *****"  
skip 3 lines  
print column 43, name clipped, " COMMODITY LINE INDEX IN ALPHABETICAL ORDER"  
skip 3 lines  
if pro = "Y" then print 38 spaces; else print 48 spaces;  
print "COMMODITY LINE", 10 spaces, "SUMMARY PAGE";  
if pro = "Y" then print 7 spaces, "FUNDING PROFILE" else print ""  
skip 2 lines
```

Jul 24 13:54 1986 comindex Page 2

```
page trailer
  skip 2 lines
  print column 47,"***** U N C L A S S I F I E D *****"
  print column 100,date,2 spaces,time
  if pletter="Z" then print 60 spaces,pno
  else print 60 spaces,pletter,"-",pno using "###"
  let pno = pno+1

end
```


(Command to Commodity Line to System to Funding Summary)

(last changed 2/6/86 to add mission area parameter and date-time stamp
2.7.86 add page letter parameter, change major system to commodity.
5/6/86 changed to the mamp database.
5/28/86 deleted command sort, now only one or more sheets per commodity.
6/5/86 added acquisition strategy from new com file and sort the systems by
acquisition code.
6/6/86 added system pdip at the expense of two outyears ***'s.
6/9/86 changed pdip to the highest priority pdip.
7/29/86 changed to unclassified since the deficiency numbers are no longer
classified.)

database mamp end

define

variable cnt type integer
variable head type integer
variable cont type integer
variable break type integer
variable obreak type integer
variable pno type integer
param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12

end

input

prompt for pno using "Please enter the starting page number > "
end

output

left margin 0
right margin 132
report to "commod.out"
end

read into b

scl_ssn
where scl_name = name and scl_code = code
end

read into a

b
ssn_acq_code ssn_amc_mgr ssn_com_line ssn_tradoc_ma ssn_da_ma
ssndroll
hipri_pdip
lrrdp_title
lrrdp_procf1 lrrdp_procf2 lrrdp_procf3 lrrdp_procf4
lrrdp_procf5 lrrdp_procf6 lrrdp_procf7 lrrdp_procf8
lrrdp_procf9 lrrdp_procf10 lrrdp_procf11 lrrdp_procf12
lrrdp_procf13 lrrdp_procf14
joining b.scl_ssn = optional ssn_ssn
and b.scl_ssn = optional ssnd_ssn
and b.scl_ssn = optional lrrdp_ssn
and b.scl_ssn = optional hipri_ssn

```
end

read into c
  a
  com_acq_strat
  acq_type
  joining a.ssn_com_line = optional com_com_line
         and a.ssn_acq_code = acq_code
end

sort by ssn_com_line ssn_acq_code scl_ssn end

format

page header
  let cnt = 0
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 46;
  if ssn_com_line <> " " then print ssn_com_line clipped;
                        else print "_____";
  print " COMMODITY LINE SUMMARY FOR ", name clipped;
  if cont = 1 then print " (continued)" else print ""
  let cont = 0
  skip 2 lines
  let head = 1

page trailer
  print "Legend:", column 47, "***** U N C L A S S I F I E D *****"
  print "   *** - Funded Procurement", column 100, date, 2 spaces, time
  if pletter="Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1

before group of ssn_com_line
  skip to top of page
  print "Acquisition Strategy: ";
  let break = 110
  while com_acq_strat[break, break] <> " " do let break = break-1
  print com_acq_strat[1, break]
  let obreak = break+1
  let break = break+125
  while com_acq_strat[break, break] <> " " do let break = break-1
  print 7 spaces, com_acq_strat[obreak, break]
  let obreak = break+1
  let break = break+125
  while com_acq_strat[break, break] <> " " do let break = break-1
  print 7 spaces, com_acq_strat[obreak, break]
  let obreak = break+1
  let break = break+125
  while com_acq_strat[break, break] <> " " do let break = break-1
  print 7 spaces, com_acq_strat[obreak, break]
  let obreak = break+1
  let break = break+125
  if break > 600 then let break = 600
  while com_acq_strat[break, break] <> " " do let break = break-1
```

```
print 7 spaces, com_acq_strat[obreak, break]
skip 1 lines
let cnt = cnt+3

after group of ssn_com_line
print "-----",
"-----"

before group of ssn_acq_code
if cnt > 19 then begin
  let cnt = 1
  print "!", column 67, "!", column 75, "!", column 132, "!"
  print "-----",
  "-----"
  skip to top of page
  let cnt = 1
end
if head = 1 then begin
  print column 90, "PROCUREMENT SCHEDULE"
  print "-----",
  "-----"
  print "!: SSN", 10 spaces, "TITLE/DEFICIENCIES", column 57, "DA/TDC MA",
  column 67, "!: PDIP !",
  " 86 87 88 89 90 91 92 93 94 95 96 97 98 99 !"
  print "!:-----!",
  "-----!"
  let head = 0
end
print "!", column 67, "!", column 75, "!", column 132, "!"
print "!: 9 spaces, "***** ", acq_type clipped, " *****",
column 67, "!", column 75, "!", column 132, "!"
print "!", column 67, "!", column 75, "!", column 132, "!"
let cnt = cnt+2

after group of ssn_acq_code
print "!", column 67, "!", column 75, "!", column 132, "!"

before group of scl_ssn
let cnt = cnt+1
if cnt > 21 then begin
  let cnt = 1
  print "!", column 67, "!", column 75, "!", column 132, "!"
  print "-----",
  "-----"
  skip to top of page
  let cnt = 1
end
if head = 1 then begin
  print column 90, "PROCUREMENT SCHEDULE"
  print "-----",
  "-----"
  print "!: SSN", 10 spaces, "TITLE/DEFICIENCIES", column 57, "DA/TDC MA",
  column 67, "!: PDIP !",
  " 86 87 88 89 90 91 92 93 94 95 96 97 98 99 !"
  print "!:-----!",
  "-----!"
```

```

print "!",column 67,"!",column 75,"!",column 132,"!"
let head = 0
end
print "!",scl_ssn,2 spaces,lrrdp_title[1,45],1 space,
      ssn_da_ma," / ",ssn_tradoc_ma,column 67,"!",hipri_pdip[1,4],"-",
      hipri_pdip[5,6],"!"
after group of scl_ssn
if scl_ssn matches "4*" then print column 93,"< Stock Funded >";
else begin
if group total of lrrdp_procf1>0 then print "****"; else print " ";
if group total of lrrdp_procf2>0 then print "****"; else print " ";
if group total of lrrdp_procf3>0 then print "****"; else print " ";
if group total of lrrdp_procf4>0 then print "****"; else print " ";
if group total of lrrdp_procf5>0 then print "****"; else print " ";
if group total of lrrdp_procf6>0 then print "****"; else print " ";
if group total of lrrdp_procf7>0 then print "****"; else print " ";
if group total of lrrdp_procf8>0 then print "****"; else print " ";
if group total of lrrdp_procf9>0 then print "****"; else print " ";
if group total of lrrdp_procf10>0 then print "****"; else print " ";
if group total of lrrdp_procf11>0 then print "****"; else print " ";
if group total of lrrdp_procf12>0 then print "****"; else print " ";
if group total of lrrdp_procf13>0 then print "****"; else print " ";
if group total of lrrdp_procf14>0 then print "****"; else print " ";
end
print column 132,"!"
print "!",8 spaces,ssn_amc_mgr[1,7];
if ssnd_def1>0 then begin
print ssnd_def1 using "####";
if ssnd_con1<>" " then print "- ",ssnd_con1; else print " "; end
if ssnd_def2>0 then begin
print ssnd_def2 using "####";
if ssnd_con2<>" " then print "- ",ssnd_con2; else print " "; end
if ssnd_def3>0 then begin
print ssnd_def3 using "####";
if ssnd_con3<>" " then print "- ",ssnd_con3; else print " "; end
if ssnd_def4>0 then begin
print ssnd_def4 using "####";
if ssnd_con4<>" " then print "- ",ssnd_con4; else print " "; end
if ssnd_def5>0 then begin
print ssnd_def5 using "####";
if ssnd_con5<>" " then print "- ",ssnd_con5; else print " "; end
if ssnd_def6>0 then begin
print ssnd_def6 using "####";
if ssnd_con6<>" " then print "- ",ssnd_con6; else print " "; end
if ssnd_def7>0 then begin
print ssnd_def7 using "####";
if ssnd_con7<>" " then print "- ",ssnd_con7; else print " "; end
if ssnd_def8>0 then begin
print ssnd_def8 using "####";
if ssnd_con8<>" " then print "- ",ssnd_con8; else print " "; end
end
print column 67,"!",column 75,"!",column 132,"!"
end

```

{ Commodity Profile Report Generator part 1. Totals the funded and unfunded rdte workpackages appropriate to each specific commodity line. }
{ last modified 2/7/86. added mission area parameter and changes to ssndesc. modified 5/28/86. changed to mamp database.
7/21/86 changed to unique workpackage linkage to systems.
added parameter to indicate a specific commodity line.
deleted 85 and added 93.
7/25/86 changed to generic years and fundpro file structure. }

database mamp end

```
define
  param[1]   code           type integer
  param[2]   name          type character length 12
  param[3]   com           type character length 10
  variable   v             type character length 1
  variable   f0_tb        type long
  variable   u0_tb        type long
  variable   f0_dev       type long
  variable   u0_dev       type long
  variable   f1_tb        type long
  variable   u1_tb        type long
  variable   f1_dev       type long
  variable   u1_dev       type long
  variable   f2_tb        type long
  variable   u2_tb        type long
  variable   f2_dev       type long
  variable   u2_dev       type long
  variable   f3_tb        type long
  variable   u3_tb        type long
  variable   f3_dev       type long
  variable   u3_dev       type long
  variable   f4_tb        type long
  variable   u4_tb        type long
  variable   f4_dev       type long
  variable   u4_dev       type long
  variable   f5_tb        type long
  variable   u5_tb        type long
  variable   f5_dev       type long
  variable   u5_dev       type long
  variable   f6_tb        type long
  variable   u6_tb        type long
  variable   f6_dev       type long
  variable   u6_dev       type long
  variable   f7_tb        type long
  variable   u7_tb        type long
  variable   f7_dev       type long
  variable   u7_dev       type long
end
```

```
output
  top margin 0
  left margin 0
  page length 32000
  report to "compr1.out"
```

```
end

read into a
  scl_ssn
  where scl_code = code and scl_name = name
end

read into b
  a
  ssn_com_line
  where ssn_com_line matches com
  joining a.scl_ssn = ssn_ssn
end

read into c
  b
  wkpkg_no_idx wkpkg_subcat
  wkpfr_f0 wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
  wkpfr_u0 wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
  joining b.scl_ssn = uniwkp_ssn
  and uniwkp_wkpkg_idx = wkpkg_wkpsys_idx
  and wkpkg_no_idx = wkpfr_wkpkg_idx
end

sort by ssn_com_line wkpkg_cmd wkpkg_cat wkpkg_no end

format

before group of ssn_com_line
  let f0_tb = 0
  let u0_tb = 0
  let f0_dev = 0
  let u0_dev = 0
  let f1_tb = 0
  let u1_tb = 0
  let f1_dev = 0
  let u1_dev = 0
  let f2_tb = 0
  let u2_tb = 0
  let f2_dev = 0
  let u2_dev = 0
  let f3_tb = 0
  let u3_tb = 0
  let f3_dev = 0
  let u3_dev = 0
  let f4_tb = 0
  let u4_tb = 0
  let f4_dev = 0
  let u4_dev = 0
  let f5_tb = 0
  let u5_tb = 0
  let f5_dev = 0
  let u5_dev = 0
  let f6_tb = 0
  let u6_tb = 0
  let f6_dev = 0
```

```
let u6_dev = 0
let f7_tb = 0
let u7_tb = 0
let f7_dev = 0
let u7_dev = 0
```

before group of wkpkg_no

```
if wkpkg_subcat="6.1" or wkpkg_subcat="6.2" or wkpkg_subcat="6.3A" then begin
```

```
let f0_tb = f0_tb + wkpfr_f0
let f1_tb = f1_tb + wkpfr_f1
let f2_tb = f2_tb + wkpfr_f2
let f3_tb = f3_tb + wkpfr_f3
let f4_tb = f4_tb + wkpfr_f4
let f5_tb = f5_tb + wkpfr_f5
let f6_tb = f6_tb + wkpfr_f6
let f7_tb = f7_tb + wkpfr_f7
let u0_tb = u0_tb + wkpfr_u0
let u1_tb = u1_tb + wkpfr_u1
let u2_tb = u2_tb + wkpfr_u2
let u3_tb = u3_tb + wkpfr_u3
let u4_tb = u4_tb + wkpfr_u4
let u5_tb = u5_tb + wkpfr_u5
let u6_tb = u6_tb + wkpfr_u6
let u7_tb = u7_tb + wkpfr_u7
```

end

```
else begin
```

```
let f0_dev = f0_dev + wkpfr_f0
let f1_dev = f1_dev + wkpfr_f1
let f2_dev = f2_dev + wkpfr_f2
let f3_dev = f3_dev + wkpfr_f3
let f4_dev = f4_dev + wkpfr_f4
let f5_dev = f5_dev + wkpfr_f5
let f6_dev = f6_dev + wkpfr_f6
let f7_dev = f7_dev + wkpfr_f7
let u0_dev = u0_dev + wkpfr_u0
let u1_dev = u1_dev + wkpfr_u1
let u2_dev = u2_dev + wkpfr_u2
let u3_dev = u3_dev + wkpfr_u3
let u4_dev = u4_dev + wkpfr_u4
let u5_dev = u5_dev + wkpfr_u5
let u6_dev = u6_dev + wkpfr_u6
let u7_dev = u7_dev + wkpfr_u7
```

end

after group of ssn_com_line

```
let v="!"
print "0", v, ssn_com_line, v, " ", v;
print f0_tb, v, f1_tb, v, f2_tb, v, f3_tb, v, f4_tb, v, f5_tb, v, f6_tb, v, f7_tb, v;
print u0_tb, v, u1_tb, v, u2_tb, v, u3_tb, v, u4_tb, v, u5_tb, v, u6_tb, v, u7_tb, v;
print f0_dev, v, f1_dev, v, f2_dev, v, f3_dev, v, f4_dev, v, f5_dev, v, f6_dev, v, f7_dev, v;
print u0_dev, v, u1_dev, v, u2_dev, v, u3_dev, v, u4_dev, v, u5_dev, v, u6_dev, v, u7_dev, v
```

end

(Commodity Profile Report Generator)

(last modified 2/7/86. added page letter and mission area parameters, and
date-time stamp on output and changes to ssndesc.
modified 5/28/86. changed to mamp database.
7/21/86. changed fy85 to fy01. added commodity parameter.
7/26/86. changed to generic rdte funding rollup file (from compro1).)

database mamp end

define

param[1]	pletter	type character	length 1
param[2]	code	type integer	
param[3]	name	type character	length 12
param[4]	com	type character	length 10
variable	pno	type integer	
variable	counter	type integer	
variable	tf	type float	
variable	i	type integer	
variable	x	type long	
variable	xdiv	type long	
variable	form1	type character	length 10
variable	form2	type character	length 10
variable	ufx	type character	length 3
variable	ufd	type character	length 3
variable	ufp	type character	length 3
variable	fx	type character	length 3
variable	fd	type character	length 3
variable	fp	type character	length 3

variable	f1_proc	type long	
variable	f2_proc	type long	
variable	f3_proc	type long	
variable	f4_proc	type long	
variable	f5_proc	type long	
variable	f6_proc	type long	
variable	f7_proc	type long	
variable	f8_proc	type long	
variable	f9_proc	type long	
variable	f10_proc	type long	
variable	f11_proc	type long	
variable	f12_proc	type long	
variable	f13_proc	type long	
variable	f14_proc	type long	
variable	f15_proc	type long	
variable	f16_proc	type long	

variable	u1_proc	type long	
variable	u2_proc	type long	
variable	u3_proc	type long	
variable	u4_proc	type long	
variable	u5_proc	type long	
variable	u6_proc	type long	
variable	u7_proc	type long	
variable	u8_proc	type long	
variable	u9_proc	type long	

AD-A172 652

CSS/EMM/SOF (COMBAT SERVICE SUPPORT/ENGINEERING AND
MINE WARFARE/SPECIAL (U) MCLEAN RESEARCH CENTER INC VA
G N ROMSTEDT SEP 86 DAAK78-84-D-0032

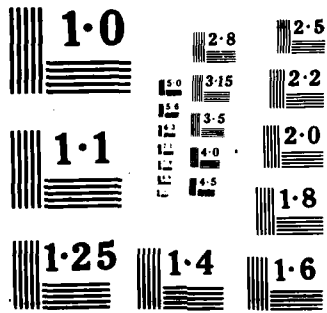
3/4

UNCLASSIFIED

F/G 9/2

NL





Jul 25 12:37 1986 compro2 Page 2

```
variable u10_proc type long
variable u11_proc type long
variable u12_proc type long
variable u13_proc type long
variable u14_proc type long
variable u15_proc type long
variable u16_proc type long
```

end

```
input
  prompt for pno using "Please enter the starting page number > "
end
```

```
output
  left margin 0
  right margin 132
  report to "compro2.out"
end
```

```
read into c
  scl_ssn
  where scl_code = code and scl_name = name
end
```

```
read into d
  c
  ssn_com_line
  where ssn_com_line matches com
  joining c.scl_ssn = ssn_ssn
end
```

```
read into a
  d
  fundpro
  joining d.ssn_com_line = optional fp_commodity
end
```

```
read into b
  a
  lrrdp_proc1 lrrdp_proc2 lrrdp_proc3 lrrdp_proc4 lrrdp_proc5 lrrdp_proc6
  lrrdp_proc7 lrrdp_proc8 lrrdp_proc9 lrrdp_proc10 lrrdp_proc11
  lrrdp_proc12 lrrdp_proc13 lrrdp_proc14 lrrdp_proc15 lrrdp_proc16
  lrrdppri_pri
  joining a.scl_ssn = optional lrrdp_ssn
  and lrrdp_pdip = optional lrrdppri_pdip
end
```

```
sort by ssn_com_line scl_ssn end
```

```
format
```

```
page header
  print column 47, "***** UNCLASSIFIED *****"
  skip 2 lines
```

```
page trailer
skip 1 line
print column 47, "***** U N C L A S S I F I E D *****"
print column 100, date, 2 spaces, time
if pletter = "Z" then print column 62, pno
else print column 62, pletter, "-", pno using "###"
let pno = pno+1

before group of ssn_com_line
skip to top of page
print 40 spaces, ssn_com_line clipped,
" COMMODITY LINE FISCAL SUMMARY FOR ", name
skip 1 line

let f1_proc = 0
let f2_proc = 0
let f3_proc = 0
let f4_proc = 0
let f5_proc = 0
let f6_proc = 0
let f7_proc = 0
let f8_proc = 0
let f9_proc = 0
let f10_proc = 0
let f11_proc = 0
let f12_proc = 0
let f13_proc = 0
let f14_proc = 0
let f15_proc = 0
let f16_proc = 0

let u1_proc = 0
let u2_proc = 0
let u3_proc = 0
let u4_proc = 0
let u5_proc = 0
let u6_proc = 0
let u7_proc = 0
let u8_proc = 0
let u9_proc = 0
let u10_proc = 0
let u11_proc = 0
let u12_proc = 0
let u13_proc = 0
let u14_proc = 0
let u15_proc = 0
let u16_proc = 0

after group of scl_ssn
if lrrdppri_pri < 1.0 or lrrdppri_pri > 800.0 then begin
let u1_proc = u1_proc+lrrdp_procf1
let u2_proc = u2_proc+lrrdp_procf2
let u3_proc = u3_proc+lrrdp_procf3
let u4_proc = u4_proc+lrrdp_procf4
let u5_proc = u5_proc+lrrdp_procf5
let u6_proc = u6_proc+lrrdp_procf6
```

```
let u7_proc = u7_proc+lrrdp_procf7
let u8_proc = u8_proc+lrrdp_procf8
let u9_proc = u9_proc+lrrdp_procf9
let u10_proc = u10_proc+lrrdp_procf10
let u11_proc = u11_proc+lrrdp_procf11
let u12_proc = u12_proc+lrrdp_procf12
let u13_proc = u13_proc+lrrdp_procf13
let u14_proc = u14_proc+lrrdp_procf14
let u15_proc = u15_proc+lrrdp_procf15
let u16_proc = u16_proc+lrrdp_procf16
end
else begin
let f1_proc = f1_proc+lrrdp_procf1
let f2_proc = f2_proc+lrrdp_procf2
let f3_proc = f3_proc+lrrdp_procf3
let f4_proc = f4_proc+lrrdp_procf4
let f5_proc = f5_proc+lrrdp_procf5
let f6_proc = f6_proc+lrrdp_procf6
let f7_proc = f7_proc+lrrdp_procf7
let f8_proc = f8_proc+lrrdp_procf8
let f9_proc = f9_proc+lrrdp_procf9
let f10_proc = f10_proc+lrrdp_procf10
let f11_proc = f11_proc+lrrdp_procf11
let f12_proc = f12_proc+lrrdp_procf12
let f13_proc = f13_proc+lrrdp_procf13
let f14_proc = f14_proc+lrrdp_procf14
let f15_proc = f15_proc+lrrdp_procf15
let f16_proc = f16_proc+lrrdp_procf16
end

after group of ssn_com_line
let form1 = "#####K "
let form2 = "(((((((#)"
let ufx = "X--"
let ufd = "D--"
let ufp = "P--"
let fx = "XXX"
let fd = "DDD"
let fp = "PPP"
let tf = 0

if f0_tb+u0_tb > tf then let tf = f0_tb+u0_tb
if f1_tb+u1_tb > tf then let tf = f1_tb+u1_tb
if f2_tb+u2_tb > tf then let tf = f2_tb+u2_tb
if f3_tb+u3_tb > tf then let tf = f3_tb+u3_tb
if f4_tb+u4_tb > tf then let tf = f4_tb+u4_tb
if f5_tb+u5_tb > tf then let tf = f5_tb+u5_tb
if f6_tb+u6_tb > tf then let tf = f6_tb+u6_tb
if f7_tb+u7_tb > tf then let tf = f7_tb+u7_tb

if f0_dev+u0_dev > tf then let tf = f0_dev+u0_dev
if f1_dev+u1_dev > tf then let tf = f1_dev+u1_dev
if f2_dev+u2_dev > tf then let tf = f2_dev+u2_dev
if f3_dev+u3_dev > tf then let tf = f3_dev+u3_dev
if f4_dev+u4_dev > tf then let tf = f4_dev+u4_dev
if f5_dev+u5_dev > tf then let tf = f5_dev+u5_dev
```

```

if f6_dev+u6_dev > tf then let tf = f6_dev+u6_dev
if f7_dev+u7_dev > tf then let tf = f7_dev+u7_dev

if (f1_proc+u1_proc)/10 > tf then let tf = (f1_proc+u1_proc)/10
if (f2_proc+u2_proc)/10 > tf then let tf = (f2_proc+u2_proc)/10
if (f3_proc+u3_proc)/10 > tf then let tf = (f3_proc+u3_proc)/10
if (f4_proc+u4_proc)/10 > tf then let tf = (f4_proc+u4_proc)/10
if (f5_proc+u5_proc)/10 > tf then let tf = (f5_proc+u5_proc)/10
if (f6_proc+u6_proc)/10 > tf then let tf = (f6_proc+u6_proc)/10
if (f7_proc+u7_proc)/10 > tf then let tf = (f7_proc+u7_proc)/10
if (f8_proc+u8_proc)/10 > tf then let tf = (f8_proc+u8_proc)/10
if (f9_proc+u9_proc)/10 > tf then let tf = (f9_proc+u9_proc)/10
if (f10_proc+u10_proc)/10 > tf then let tf = (f10_proc+u10_proc)/10
if (f11_proc+u11_proc)/10 > tf then let tf = (f11_proc+u11_proc)/10
if (f12_proc+u12_proc)/10 > tf then let tf = (f12_proc+u12_proc)/10
if (f13_proc+u13_proc)/10 > tf then let tf = (f13_proc+u13_proc)/10
if (f14_proc+u14_proc)/10 > tf then let tf = (f14_proc+u14_proc)/10
if (f15_proc+u15_proc)/10 > tf then let tf = (f15_proc+u15_proc)/10
if (f16_proc+u16_proc)/10 > tf then let tf = (f16_proc+u16_proc)/10

let tf = tf*0.8
skip 3 lines
let x = 15000
if tf>x then let x = 30000
if tf>x then let x = 75000
if tf>x then let x = 150000
if tf>x then let x = 300000
if tf>x then let x = 750000
if tf>x then let x = 1500000
let xdiv = x/30
for i = 1 to 30 do begin
  if i=1 or i=11 or i=21 then print 4 spaces, x/1000 using "#####";
  if i=15 then print "RDTE ($ 1M)";
  if i=17 then print "PROC ($10M)";
  print column 13, "!";
  if f0_tb + u0_tb >= x then begin
    if f0_tb >= x then print fx; else print ufx; end
    else print " ";
  if f0_dev + u0_dev >= x then begin
    if f0_dev >= x then print fd; else print ufd; end
    else print " ";
  if f1_proc+u1_proc >= x*10 then begin
    if f1_proc >= x*10 then print fp; else print ufp; end
    else print " ";
  print 1 space;
  if f1_tb + u1_tb >= x then begin
    if f1_tb >= x then print fx; else print ufx; end
    else print " ";
  if f1_dev + u1_dev >= x then begin
    if f1_dev >= x then print fd; else print ufd; end
    else print " ";
  if f2_proc+u2_proc >= x*10 then begin
    if f2_proc >= x*10 then print fp; else print ufp; end
    else print " ";
  print 1 space;
  if f2_tb + u2_tb >= x then begin

```

```
    if f2_tb >= x then print fx; else print ufx; end
    else print " ";
  if f2_dev + u2_dev >= x then begin
    if f2_dev >= x then print fd; else print ufd; end
    else print " ";
  if f3_proc+u3_proc >= x*10 then begin
    if f3_proc >= x*10 then print fp; else print ufp; end
    else print " ";
  print 1 space;
  if f3_tb + u3_tb >= x then begin
    if f3_tb >= x then print fx; else print ufx; end
    else print " ";
  if f3_dev + u3_dev >= x then begin
    if f3_dev >= x then print fd; else print ufd; end
    else print " ";
  if f4_proc+u4_proc >= x*10 then begin
    if f4_proc >= x*10 then print fp; else print ufp; end
    else print " ";
  print 1 space;
  if f4_tb + u4_tb >= x then begin
    if f4_tb >= x then print fx; else print ufx; end
    else print " ";
  if f4_dev + u4_dev >= x then begin
    if f4_dev >= x then print fd; else print ufd; end
    else print " ";
  if f5_proc+u5_proc >= x*10 then begin
    if f5_proc >= x*10 then print fp; else print ufp; end
    else print " ";
  print 1 space;
  if f5_tb + u5_tb >= x then begin
    if f5_tb >= x then print fx; else print ufx; end
    else print " ";
  if f5_dev + u5_dev >= x then begin
    if f5_dev >= x then print fd; else print ufd; end
    else print " ";
  if f6_proc+u6_proc >= x*10 then begin
    if f6_proc >= x*10 then print fp; else print ufp; end
    else print " ";
  print 1 space;
  if f6_tb + u6_tb >= x then begin
    if f6_tb >= x then print fx; else print ufx; end
    else print " ";
  if f6_dev + u6_dev >= x then begin
    if f6_dev >= x then print fd; else print ufd; end
    else print " ";
  if f7_proc+u7_proc >= x*10 then begin
    if f7_proc >= x*10 then print fp; else print ufp; end
    else print " ";
  print 1 space;
  if f7_tb + u7_tb >= x then begin
    if f7_tb >= x then print fx; else print ufx; end
    else print " ";
  if f7_dev + u7_dev >= x then begin
    if f7_dev >= x then print fd; else print ufd; end
    else print " ";
  if f8_proc+u8_proc >= x*10 then begin
```

```

    if f8_proc >= x*10 then print fp; else print ufp; end
    else print " ";
print 2 spaces;
if f9_proc+u9_proc >= x*10 then begin
    if f9_proc >= x*10 then print fp; else print ufp; end
    else print " ";
print 2 spaces;
if f10_proc+u10_proc >= x*10 then begin
    if f10_proc >= x*10 then print fp; else print ufp; end
    else print " ";
print 2 spaces;
if f11_proc+u11_proc >= x*10 then begin
    if f11_proc >= x*10 then print fp; else print ufp; end
    else print " ";
print 2 spaces;
if f12_proc+u12_proc >= x*10 then begin
    if f12_proc >= x*10 then print fp; else print ufp; end
    else print " ";
print 2 spaces;
if f13_proc+u13_proc >= x*10 then begin
    if f13_proc >= x*10 then print fp; else print ufp; end
    else print " ";
print 2 spaces;
if f14_proc+u14_proc >= x*10 then begin
    if f14_proc >= x*10 then print fp; else print ufp; end
    else print " ";
print 2 spaces;
if f15_proc+u15_proc >= x*10 then begin
    if f15_proc >= x*10 then print fp; else print ufp; end
    else print " ";
print 2 spaces;
if f16_proc+u16_proc >= x*10 then begin
    if f16_proc >= x*10 then print fp; else print ufp; end
    else print " ";
print ""
let x = x-xdiv
end
print 13 spaces;
for i = 14 to 132 do print "-";
print ""
print 16 spaces, "1986", 6 spaces, "1987", 6 spaces, "1988", 6 spaces,
"1989", 6 spaces, "1990", 6 spaces, "1991", 6 spaces, "1992", 6 spaces, "1993",
" : 94 95 96 97 98 99 00 01"
print column 94, "!"
print "Tech Base ", f0_tb using form1, f1_tb using form1,
f2_tb using form1, f3_tb using form1, f4_tb using form1,
f5_tb using form1, f6_tb using form1, f7_tb using form1, column 94, "!"
print " (, fx, "/", ufx, " ) ", -u0_tb using form2, -u1_tb using form2,
-u2_tb using form2, -u3_tb using form2, -u4_tb using form2,
-u5_tb using form2, -u6_tb using form2, -u7_tb using form2, column 94, "!"
print column 94, "!"
print "Development ", f0_dev using form1, f1_dev using form1,
f2_dev using form1, f3_dev using form1, f4_dev using form1,
f5_dev using form1, f6_dev using form1, f7_dev using form1, column 94, "!"
print " (, fd, "/", ufd, " ) ", -u0_dev using form2, -u1_dev using form2,
-u2_dev using form2, -u3_dev using form2, -u4_dev using form2,

```



```
-u5_dev using form2,-u6_dev using form2,-u7_dev using form2,column 94,":"  
print column 94,":"  
print "Procurement ",f1_proc using form1,f2_proc using form1,  
f3_proc using form1,f4_proc using form1,f5_proc using form1,  
f6_proc using form1,f7_proc using form1,f8_proc using form1,  
column 94,": See Out Year Procurement"  
print " (",fp,"/",ufp,") ",-u1_proc using form2,-u2_proc using form2,  
-u3_proc using form2,-u4_proc using form2,-u5_proc using form2,  
-u6_proc using form2,-u7_proc using form2,-u8_proc using form2,  
column 94,": "  
print 54 spaces;  
for i=93 to 132 do print "-";  
print ""  
print 56 spaces,"1994",6 spaces,"1995",6 spaces,"1996",6 spaces,"1997",  
6 spaces,"1998",6 spaces,"1999",6 spaces,"2000",6 spaces,"2001"  
skip 1 line  
print 29 spaces,"Procurement (Out Years)",f9_proc using form1,  
f10_proc using form1,f11_proc using form1,f12_proc using form1,  
f13_proc using form1,f14_proc using form1,f15_proc using form1,  
f16_proc using form1  
print 52 spaces,-u9_proc using form2,-u10_proc using form2,  
-u11_proc using form2,-u12_proc using form2,-u13_proc using form2,  
-u14_proc using form2,-u15_proc using form2,-u16_proc using form2  
  
end
```

< 6/25/86. COMMODITY/SYSTEM/PDIP report.
made for Bob Brown of the Engineer School to show all the funding data in
the data base for each system in the control file. >

database mamp end

```
define
  variable cont      type integer
  variable cnt       type integer
  variable pno       type integer
  param[1] pletter  type character length 1
  param[2] code      type integer
  param[3] name      type character length 12
end
```

```
input
  prompt for pno using "Please enter the starting page number > "
end
```

```
output
  left margin 0
  right margin 132
  report to "comsyspdip.out"
end
```

```
read into b
  scl_ssn
  where scl_name = name and scl_code = code
end
```

```
read into c
  b
  ssn_amc_mgr ssn_acq_code ssn_com_line ssn_tradoc_ma ssn_da_ma
  acq_type
  joining b. scl_ssn = optional ssn_ssn
  and ssn_acq_code = optional acq_code
end
```

```
read into a
  c
  lrrdp_title lrrdp_pdip
  lrrdp_procf1 lrrdp_procf2 lrrdp_procf3 lrrdp_procf4 lrrdp_procf5
  lrrdp_procf6 lrrdp_procf7 lrrdp_procf8 lrrdp_procf9 lrrdp_procf10
  lrrdp_procf11 lrrdp_procf12 lrrdp_procf13 lrrdp_procf14 lrrdp_procf15
  lrrdp_procf16
  joining c. scl_ssn = optional lrrdp_ssn
end
```

```
sort by ssn_com_line ssn_acq_code scl_ssn lrrdp_pdip end
```

```
format
```

```
page header
  print column 47, "***** UNCLASSIFIED *****"
  skip 2 lines
  print column 44;
```

```

if ssn_com_line <> " " then print ssn_com_line clipped;
                        else print "_____";
print " COMMODITY/SYSTEM/PDIP SUMMARY FOR ",name clipped;
if cont = 1 then print " (continued)" else print ""
let cont = 0
skip 2 lines

page trailer
skip 2 lines
print column 47,"***** U N C L A S S I F I E D *****"
print column 100,date,2 spaces,time
if pletter="Z" then print column 60,pno
else print column 60,pletter,"-",pno using "###"
let pno = pno+1

before group of ssn_com_line
skip to top of page

after group of ssn_com_line
let cont = 0

before group of scl_ssn
let cont = 1
let cnt = 0
skip 1 line
print scl_ssn,2 spaces,lrrdp_title,1 space,
      ssn_da_ma," / ",ssn_tradoc_ma,3 spaces,ssn_amc_mgr
print 8 spaces," PDIP ",5 spaces,
      "   fy87   fy88   fy89   fy90   fy91   fy92   fy93   fy94",
      "   fy95   fy96   fy97   fy98   fy99   fy00   fy01"

before group of lrrdp_pdip
let cnt = cnt+1
if lrrdp_pdip <> " " then print 8 spaces,lrrdp_pdip,5 spaces;
else print 8 spaces,"N/A ",5 spaces;
print lrrdp_procf2 using " #####", lrrdp_procf3 using " #####",
lrrdp_procf4 using " #####", lrrdp_procf5 using " #####",
lrrdp_procf6 using " #####", lrrdp_procf7 using " #####",
lrrdp_procf8 using " #####", lrrdp_procf9 using " #####",
lrrdp_procf10 using " #####", lrrdp_procf11 using " #####",
lrrdp_procf12 using " #####", lrrdp_procf13 using " #####",
lrrdp_procf14 using " #####", lrrdp_procf15 using " #####",
lrrdp_procf16 using " #####"

after group of scl_ssn
if cnt > 1 then print 8 spaces,"Total:",5 spaces,
group total of lrrdp_procf2 using " #####",
group total of lrrdp_procf3 using " #####",
group total of lrrdp_procf4 using " #####",
group total of lrrdp_procf5 using " #####",
group total of lrrdp_procf6 using " #####",
group total of lrrdp_procf7 using " #####",
group total of lrrdp_procf8 using " #####",
group total of lrrdp_procf9 using " #####",
group total of lrrdp_procf10 using " #####",
group total of lrrdp_procf11 using " #####",

```

Jun 25 10:45 1986 comsyspdip Page 3

```
group total of lrrdp_procf12 using " *****",  
group total of lrrdp_procf13 using " *****",  
group total of lrrdp_procf14 using " *****",  
group total of lrrdp_procf15 using " *****",  
group total of lrrdp_procf16 using " *****"
```

```
before group of ssn_acq_code  
skip 1 line  
print 39 spaces,"***** ",acq_type clipped," *****"
```

```
after group of ssn_acq_code  
skip 1 line
```

end

(DECISION AID FOR SYSTEMS BASED ON BDP PRIORITIES.
THIS REPORT SHOWS THE UNFUNDED WORKPACKAGES THAT HAVE A RATING OF
1, 2, OR 3 AGAINST EACH SYSTEM.

4/25/86 changed to mamp data base.

5/12/86 added base case systems and put them at the top of the list.

7/26/86 changed wkpsys_wc to wkpsys_srf in keeping with iitri's usage.
changed base year to 1986.)

database mamp end

define

variable pno type integer
variable pgend type integer
variable cnt type integer
variable totuf type long
param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12

end

input

prompt for pno using "Please enter the starting page number > "
end

output

left margin 0
right margin 132
report to "decsysbdp.out"

end

read into b

scl_ssn
where scl_code = code and scl_name = name
end

read into c

b
ssn_amc_mgr ssn_da_ma ssn_tradoc_ma
bc = 0
where ssn_acq_code > 2 or ssn_acq_code = 0
joining b.scl_ssn = optional ssn_ssn
end

read into d

b
ssn_amc_mgr ssn_da_ma ssn_tradoc_ma
bc = 1
where ssn_acq_code = 1
joining b.scl_ssn = optional ssn_ssn
end

assign e = c union d end

read into a

e

```
hipri_title
priori_score priori_ndef
joining e scl_ssn = optional hipri_ssn
      and e.scl_ssn = optional priori_ssn_no
end

read into c
  b
  wkpsys_srf wkpkg_wkpsys_idx wkpkg_title
  wkpf_yr wkpf_unfund
  where wkpf_unfund > 0 and wkpsys_srf > 0 and wkpsys_srf < 4
  joining b.scl_ssn = wkpsys_ssn
        and wkpsys_wkpkg_idx = wkpkg_wkpsys_idx
        and wkpkg_no_idx = wkpf_wkpkg_idx
end

read into d
  a
  c.wkpsys_srf c.wkpkg_wkpsys_idx c.wkpkg_title c.wkpf_yr c.wkpf_unfund
  joining a.scl_ssn = optional c.scl_ssn
end

sort by bc descending priori_score descending priori_ndef descending
      scl_ssn wkpsys_srf wkpkg_no wkpf_yr end

format

page header
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 44, name clipped, " 1 TO N DEVELOPMENT SYSTEMS BDP RANKINGS"
  print column 44, "WITH SIGNIFICANT UNFUNDED WORKPACKAGES SHOWN"
  skip 3 lines
  print column 86, "UNFUNDED DEVELOPMENT SCHEDULE"
  print "-----",
  "-----"
  print ":", SSN", 5 spaces, "TITLE / UNFUNDED WORKPACKAGES",
  column 57, "DA/TDC MA", column 67,
  ":", FY86 FY87 FY88 FY89 FY90 FY91 FY92 FY93 ":"
  print ":",
  ":",
  print ":", column 67, ":", column 132, ":",
  let pgend = 0

page trailer
  if pgend = 0 then begin
    print ":", column 67, ":", column 132, ":",
    print "-----",
    "-----"
  end
  else skip 2 lines
  skip 2 lines
  print column 47, "***** U N C L A S S I F I E D *****"
  print column 100, date, 2 spaces, time
  if pletter = "Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
```

```
    let pno = pno+1

on last record
  let pgend = 1
  print "!", column 67, "!", column 132, "!"
  print "-----"
  print "-----"

before group of scl_ssn
  let cnt = cnt+1
  print "!", cnt using "##.", scl_ssn, 2 spaces, htpri_title[1, 40],
    1 space, ssn_da_ma, " / ", ssn_tradoc_ma,
    column 67, "!", column 132, "!"

before group of wkpkg_no
  if wkpf_yr > 1985 then begin
    print "!", 8 spaces, wkpkg_cmd clipped, 1 space, wkpkg_pe, 1 space,
      wkpkg_proj, 1 space, wkpkg_task clipped, 1 space, wkpkg_no clipped,
      column 48, "crit: ", wkpsys_srf using "#", column 67, "!"
  end

before group of wkpf_yr
  if wkpf_yr > 1985 then begin
    if wkpf_unfund > 0 then print column (8*(wkpf_yr-1986)+68),
      -wkpf_unfund using "((((((#)";
  end

after group of wkpkg_no
  if wkpf_yr > 1985 then begin
    print column 132, "!"
    print "!", 11 spaces, wkpkg_title[1, 52], column 67, "!", column 132, "!"
  end

end
```

< DECISION AID FOR WORKPACKAGES BASED ON BDP PRIORITIES.
THIS REPORT SHOWS THE UNFUNDED WORKPACKAGES.

4/25/86 changed to mamp data base.
5/12/86 limited to 6.3, 6.4, and 6.7 workpackages.
7/25/86 changed to 1986 as base year.)

```
database mamp end

define
  variable pno      type integer
  variable pgen     type integer
  variable cnt      type integer
  variable totuf    type long
  param[1] pletter  type character length 1
  param[2] code     type integer
  param[3] name     type character length 12
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  left margin 0
  right margin 132
  report to "decupbdb.out"
end

read into b
  pcl_idx
  where pcl_code = code and pcl_name = name
     and (pcl_subcat = "6.3A" or pcl_subcat = "6.3B" or pcl_subcat = "6.4"
        or pcl_subcat = "6.7")
end

read into c
  prior2_score prior2_nsys
  wkpkg_no_idx wkpkg_pe wkpkg_proj wkpkg_task wkpkg_title
  joining b.pcl_idx = wkpkg_proj_idx
     and wkpkg_no_idx = optional prior2_wkpkg_idx
end

read into d
  wkpf_wkpkg_idx wkpf_yr wkpf_unfund
  where wkpf_unfund > 0
  joining c.wkpkg_no_idx = wkpf_wkpkg_idx
end

read into a
  c
  d.wkpf_yr d.wkpf_unfund
  joining c.wkpkg_no_idx = optional d.wkpf_wkpkg_idx
end

sort by prior2_score descending prior2_nsys descending
```



```
      wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no wkpf_yr end

format

page header
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 48, name clipped, " 1 TO N WORKPACKAGE BDP RANKINGS"
  print column 54, "WITH UNFUNDED AMOUNTS SHOWN"
  skip 3 lines
  print column 86, "UNFUNDED DEVELOPMENT SCHEDULE"
  print "-----"
  print "
  print " |          PE/PROJ/TASK/WKPKG", 5 spaces, "TITLE", column 67,
  print " | FY86   FY87   FY88   FY89   FY90   FY91   FY92   FY93 |"
  print " |-----"
  print " |"
  print " |", column 67, " |", column 132, " |"
  let pgend = 0

page trailer
  if pgend = 0 then begin
    print " |", column 67, " |", column 132, " |"
    print "-----"
  end
  else skip 2 lines
  skip 2 lines
  print column 47, "***** U N C L A S S I F I E D *****"
  print column 100, date, 2 spaces, time
  if pletter = "Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1

on last record
  let pgend = 1
  print " |", column 67, " |", column 132, " |"
  print "-----"

before group of wkpkg_no
  let cnt = cnt+1
  print " | ", cnt using "###. ", wkpkg_title[1, 55], column 67, " |";

before group of wkpf_yr
  if wkpf_yr > 1985 then begin
    if wkpf_unfund > 0 then print column (8*(wkpf_yr-1986)+68),
      -wkpf_unfund using "((((((#)";
  end

after group of wkpkg_no
  print column 132, " |"
  print " | ", 5 spaces, wkpkg_cmd, 1 space,
  wkpkg_pe, 1 space, wkpkg_proj, 1 space, wkpkg_task clipped, 1 space,
  wkpkg_no, column 67, " |", column 132, " |"
```

Jul 25 11:16 1986 decupbdb Page 3

end

D-45

Jul 24 14:25 1986 defindex Page 1

{ Deficiency index for the rollup report only }

database mamp end

{ last modified 2/7/86 to add mission area to control file.
3/6/86 to add page letter and page number
5/12/86 changed to the mamp data base.
6/13/86 changed to always print out primary (since type is screwed up).
7/24/86 merged with defindex1 by adding profile column parameter. }

define

param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12
param[4] pro type character length 1
variable pno type integer

end

input

prompt for pno using "Please enter the starting page number > "
end

output

left margin 0
right margin 132
report to "defindex.out"
end

read into a

defctl
where dcl_code = code and dcl_name = name
end

sort by dcl_def end

format

before group of dcl_def

if pro = "Y" then print 18 spaces; else print 28 spaces;
print dcl_def, 70 spaces, "----";
if pro = "Y" then print 12 spaces, "----" else print ""

page header

print column 47, "***** U N C L A S S I F I E D *****"
skip 3 lines
print column 47, name clipped, " DEFICIENCY INDEX IN NUMERICAL ORDER"
skip 3 lines
if pro = "Y" then print 18 spaces; else print 28 spaces;
print "DEFICIENCY", 16 spaces, "TITLE", 41 spaces, "SUMMARY PAGE";
if pro = "Y" then print 4 spaces, "FUNDING PROFILE" else print ""
skip 2 lines

page trailer

skip 2 lines
print column 47, "***** U N C L A S S I F I E D *****"
print column 100, date, 2 spaces, time

Jul 24 14:25 1986 defindex Page 2

```
if pletter="Z" then print 60 spaces,pno
else print 60 spaces,pletter,"-",pno using "###"
let pno = pno+1
```

end

< Deficiency Profile Report Generator part 1 >

< last changed 2/7/86. added mission area parameter.
modified 3/28/86. changed to mamp database.
7/24/86. changed to generic years.
7/25/86. changed to generic funding rollup file. >

database mamp end

```
define
  param[1]   code           type integer
  param[2]   name          type character length 12
  variable   v             type character length 1
  variable   f0_tb         type long
  variable   u0_tb         type long
  variable   f0_dev        type long
  variable   u0_dev        type long
  variable   f1_tb         type long
  variable   u1_tb         type long
  variable   f1_dev        type long
  variable   u1_dev        type long
  variable   f2_tb         type long
  variable   u2_tb         type long
  variable   f2_dev        type long
  variable   u2_dev        type long
  variable   f3_tb         type long
  variable   u3_tb         type long
  variable   f3_dev        type long
  variable   u3_dev        type long
  variable   f4_tb         type long
  variable   u4_tb         type long
  variable   f4_dev        type long
  variable   u4_dev        type long
  variable   f5_tb         type long
  variable   u5_tb         type long
  variable   f5_dev        type long
  variable   u5_dev        type long
  variable   f6_tb         type long
  variable   u6_tb         type long
  variable   f6_dev        type long
  variable   u6_dev        type long
  variable   f7_tb         type long
  variable   u7_tb         type long
  variable   f7_dev        type long
  variable   u7_dev        type long
end
```

```
output
  top margin 0
  bottom margin 0
  left margin 0
  page length 32000
  report to "defpro1.out"
end
```

```
read into a
```

```
defctl
  where dcl_code = code and dcl_name = name
end

read into b
  a
  ssndef_ssn
  joining a.dcl_def = ssndef_def
end

read into c
  b
  wkpkg_cmd wkpkg_cat wkpkg_subcat wkpkg_no
  wkpfr_f0 wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
  wkpfr_u0 wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
  joining b.ssndef_ssn = wkpsys_ssn
  and wkpsys_wkpkg_idx = wkpkg_wkpsys_idx
  and wkpkg_no_idx = wkpfr_wkpkg_idx
end

sort by dcl_def wkpkg_cmd wkpkg_cat wkpkg_no end

format

before group of dcl_def
  let f0_tb = 0
  let u0_tb = 0
  let f0_dev = 0
  let u0_dev = 0
  let f1_tb = 0
  let u1_tb = 0
  let f1_dev = 0
  let u1_dev = 0
  let f2_tb = 0
  let u2_tb = 0
  let f2_dev = 0
  let u2_dev = 0
  let f3_tb = 0
  let u3_tb = 0
  let f3_dev = 0
  let u3_dev = 0
  let f4_tb = 0
  let u4_tb = 0
  let f4_dev = 0
  let u4_dev = 0
  let f5_tb = 0
  let u5_tb = 0
  let f5_dev = 0
  let u5_dev = 0
  let f6_tb = 0
  let u6_tb = 0
  let f6_dev = 0
  let u6_dev = 0
  let f7_tb = 0
  let u7_tb = 0
  let f7_dev = 0
```

```
let u7_dev = 0

before group of wkpkg_no
if wkpkg_subcat="6.1" or wkpkg_subcat="6.2" or wkpkg_subcat="6.3A" then begin
let f0_tb = f0_tb + wkpfr_f0
let f1_tb = f1_tb + wkpfr_f1
let f2_tb = f2_tb + wkpfr_f2
let f3_tb = f3_tb + wkpfr_f3
let f4_tb = f4_tb + wkpfr_f4
let f5_tb = f5_tb + wkpfr_f5
let f6_tb = f6_tb + wkpfr_f6
let f7_tb = f7_tb + wkpfr_f7
let u0_tb = u0_tb + wkpfr_u0
let u1_tb = u1_tb + wkpfr_u1
let u2_tb = u2_tb + wkpfr_u2
let u3_tb = u3_tb + wkpfr_u3
let u4_tb = u4_tb + wkpfr_u4
let u5_tb = u5_tb + wkpfr_u5
let u6_tb = u6_tb + wkpfr_u6
let u7_tb = u7_tb + wkpfr_u7
end
else if wkpkg_subcat="6.3B" or wkpkg_subcat="6.4" or wkpkg_subcat="6.7" then begin
let f0_dev = f0_dev + wkpfr_f0
let f1_dev = f1_dev + wkpfr_f1
let f2_dev = f2_dev + wkpfr_f2
let f3_dev = f3_dev + wkpfr_f3
let f4_dev = f4_dev + wkpfr_f4
let f5_dev = f5_dev + wkpfr_f5
let f6_dev = f6_dev + wkpfr_f6
let f7_dev = f7_dev + wkpfr_f7
let u0_dev = u0_dev + wkpfr_u0
let u1_dev = u1_dev + wkpfr_u1
let u2_dev = u2_dev + wkpfr_u2
let u3_dev = u3_dev + wkpfr_u3
let u4_dev = u4_dev + wkpfr_u4
let u5_dev = u5_dev + wkpfr_u5
let u6_dev = u6_dev + wkpfr_u6
let u7_dev = u7_dev + wkpfr_u7
end

after group of dcl_def
let v="!"
print dcl_def,v," ",v," ",v,
print f0_tb,v,f1_tb,v,f2_tb,v,f3_tb,v,f4_tb,v,f5_tb,v,f6_tb,v,f7_tb,v,
print u0_tb,v,u1_tb,v,u2_tb,v,u3_tb,v,u4_tb,v,u5_tb,v,u6_tb,v,u7_tb,v,
print f0_dev,v,f1_dev,v,f2_dev,v,f3_dev,v,f4_dev,v,f5_dev,v,f6_dev,v,f7_dev,v,
print u0_dev,v,u1_dev,v,u2_dev,v,u3_dev,v,u4_dev,v,u5_dev,v,u6_dev,v,u7_dev,v

end
```

{ Deficiency Profile Report Generator }

{ last modified 2/7/86. to add page letter parameter, mission area and
date time stamp.
5/28/86. changed to mamp database.
7/24/86. changed to generic years. made unclass.
7/25/86. changed to generic rdte rollup file. }

database mamp end

```
define
param[1]    pletter      type character length 1
param[2]    code         type integer
param[3]    name         type character length 12
variable    pno          type integer
variable    counter      type integer
variable    tf           type float
variable    i            type integer
variable    x            type long
variable    xdiv         type long
variable    form1        type character length 10
variable    form2        type character length 10
variable    ufx          type character length 3
variable    ufd          type character length 3
variable    ufp          type character length 3
variable    fx           type character length 3
variable    fd           type character length 3
variable    fp           type character length 3

variable    f1_proc      type long
variable    f2_proc      type long
variable    f3_proc      type long
variable    f4_proc      type long
variable    f5_proc      type long
variable    f6_proc      type long
variable    f7_proc      type long
variable    f8_proc      type long
variable    f9_proc      type long
variable    f10_proc     type long
variable    f11_proc     type long
variable    f12_proc     type long
variable    f13_proc     type long
variable    f14_proc     type long
variable    f15_proc     type long
variable    f16_proc     type long
variable    u1_proc      type long
variable    u2_proc      type long
variable    u3_proc      type long
variable    u4_proc      type long
variable    u5_proc      type long
variable    u6_proc      type long
variable    u7_proc      type long
variable    u8_proc      type long
variable    u9_proc      type long
variable    u10_proc     type long
variable    u11_proc     type long
```


Jul 25 12:17 1986 defpro2 Page 2

```
variable u12_proc type long
variable u13_proc type long
variable u14_proc type long
variable u15_proc type long
variable u16_proc type long
end

input
  prompt for pno using "Please enter the starting page number 0:"
end

output
  left margin 0
  right margin 132
  report to "defpro2 out"
end

read into c
  dcl_def
  where dcl_code = code and dcl_name = name
end

read into a
  c
  fundpro
  joining c.dcl_def = optional fp_defic
end

read into b
  a
  ssn_ssn ssn_tradoc_ma ssn_tradoc_pro ssn_da_ma
  joining a.dcl_def = optional ssndef_def
  and ssndef_ssn = optional ssn_ssn
end

read into c
  b
  lrrdp_pdip
  lrrdp_procf1 lrrdp_procf2 lrrdp_procf3 lrrdp_procf4 lrrdp_procf5 lrrdp_procf6
  lrrdp_procf7 lrrdp_procf8 lrrdp_procf9 lrrdp_procf10 lrrdp_procf11
  lrrdp_procf12 lrrdp_procf13 lrrdp_procf14 lrrdp_procf15 lrrdp_procf16
  lrrdpPri_pri
  joining b.ssn_ssn = optional lrrdp_ssn
  and lrrdp_pdip = optional lrrdpPri_pdip
end

sort by dcl_def ssn_ssn lrrdp_pdip end

format

page header
  print column 47, "***** UNCLASSIFIED *****"
  skip 2 lines

page trailer
  skip 1 line
```

```
print column 47, "***** U N C L A S S I F I E D *****"  
print column 100, date, 2 spaces, time  
if pletter="Z" then print column 60, pno  
else print column 60, pletter, "-", pno using "###"  
let pno = pno+1
```

```
before group of dcl_def  
skip to top of page  
print 40 spaces, "FISCAL SUMMARY - DEFICIENCY ", dcl_def using "CCCC"  
skip 1 line  
let f1_proc = 0  
let f2_proc = 0  
let f3_proc = 0  
let f4_proc = 0  
let f5_proc = 0  
let f6_proc = 0  
let f7_proc = 0  
let f8_proc = 0  
let f9_proc = 0  
let f10_proc = 0  
let f11_proc = 0  
let f12_proc = 0  
let f13_proc = 0  
let f14_proc = 0  
let f15_proc = 0  
let f16_proc = 0  
let u1_proc = 0  
let u2_proc = 0  
let u3_proc = 0  
let u4_proc = 0  
let u5_proc = 0  
let u6_proc = 0  
let u7_proc = 0  
let u8_proc = 0  
let u9_proc = 0  
let u10_proc = 0  
let u11_proc = 0  
let u12_proc = 0  
let u13_proc = 0  
let u14_proc = 0  
let u15_proc = 0  
let u16_proc = 0
```

```
before group of lrrdp_pdp  
if lrrdppri_pri < 1.0 or lrrdppri_pri > 800.0 then begin  
let u1_proc = u1_proc+lrrdp_procf1  
let u2_proc = u2_proc+lrrdp_procf2  
let u3_proc = u3_proc+lrrdp_procf3  
let u4_proc = u4_proc+lrrdp_procf4  
let u5_proc = u5_proc+lrrdp_procf5  
let u6_proc = u6_proc+lrrdp_procf6  
let u7_proc = u7_proc+lrrdp_procf7  
let u8_proc = u8_proc+lrrdp_procf8  
let u9_proc = u9_proc+lrrdp_procf9  
let u10_proc = u10_proc+lrrdp_procf10  
let u11_proc = u11_proc+lrrdp_procf11
```

```
let u12_proc = u12_proc+lrrdp_procf12
let u13_proc = u13_proc+lrrdp_procf13
let u14_proc = u14_proc+lrrdp_procf14
let u15_proc = u15_proc+lrrdp_procf15
let u16_proc = u16_proc+lrrdp_procf16
end
else begin
let f1_proc = f1_proc+lrrdp_procf1
let f2_proc = f2_proc+lrrdp_procf2
let f3_proc = f3_proc+lrrdp_procf3
let f4_proc = f4_proc+lrrdp_procf4
let f5_proc = f5_proc+lrrdp_procf5
let f6_proc = f6_proc+lrrdp_procf6
let f7_proc = f7_proc+lrrdp_procf7
let f8_proc = f8_proc+lrrdp_procf8
let f9_proc = f9_proc+lrrdp_procf9
let f10_proc = f10_proc+lrrdp_procf10
let f11_proc = f11_proc+lrrdp_procf11
let f12_proc = f12_proc+lrrdp_procf12
let f13_proc = f13_proc+lrrdp_procf13
let f14_proc = f14_proc+lrrdp_procf14
let f15_proc = f15_proc+lrrdp_procf15
let f16_proc = f16_proc+lrrdp_procf16
end
end

after group of dcl_def
let form1 = "##### "
let form2 = "(((((((#)"
let ufx = "X--"
let ufd = "D--"
let ufp = "P--"
let fx = "XXY"
let fd = "DDD"
let fp = "PPP"

let tf = 0
if f0_tb+u0_tb > tf then let tf = f0_tb+u0_tb
if f1_tb+u1_tb > tf then let tf = f1_tb+u1_tb
if f2_tb+u2_tb > tf then let tf = f2_tb+u2_tb
if f3_tb+u3_tb > tf then let tf = f3_tb+u3_tb
if f4_tb+u4_tb > tf then let tf = f4_tb+u4_tb
if f5_tb+u5_tb > tf then let tf = f5_tb+u5_tb
if f6_tb+u6_tb > tf then let tf = f6_tb+u6_tb
if f7_tb+u7_tb > tf then let tf = f7_tb+u7_tb
if f0_dev+u0_dev > tf then let tf = f0_dev+u0_dev
if f1_dev+u1_dev > tf then let tf = f1_dev+u1_dev
if f2_dev+u2_dev > tf then let tf = f2_dev+u2_dev
if f3_dev+u3_dev > tf then let tf = f3_dev+u3_dev
if f4_dev+u4_dev > tf then let tf = f4_dev+u4_dev
if f5_dev+u5_dev > tf then let tf = f5_dev+u5_dev
if f6_dev+u6_dev > tf then let tf = f6_dev+u6_dev
if f7_dev+u7_dev > tf then let tf = f7_dev+u7_dev
if (f1_proc+u1_proc)/10 > tf then let tf = (f1_proc+u1_proc)/10
if (f2_proc+u2_proc)/10 > tf then let tf = (f2_proc+u2_proc)/10
if (f3_proc+u3_proc)/10 > tf then let tf = (f3_proc+u3_proc)/10
if (f4_proc+u4_proc)/10 > tf then let tf = (f4_proc+u4_proc)/10
```

```

if (f5_proc+u5_proc)/10 > tf then let tf = (f5_proc+u5_proc)/10
if (f6_proc+u6_proc)/10 > tf then let tf = (f6_proc+u6_proc)/10
if (f7_proc+u7_proc)/10 > tf then let tf = (f7_proc+u7_proc)/10
if (f8_proc+u8_proc)/10 > tf then let tf = (f8_proc+u8_proc)/10
if (f9_proc+u9_proc)/10 > tf then let tf = (f9_proc+u9_proc)/10
if (f10_proc+u10_proc)/10 > tf then let tf = (f10_proc+u10_proc)/10
if (f11_proc+u11_proc)/10 > tf then let tf = (f11_proc+u11_proc)/10
if (f12_proc+u12_proc)/10 > tf then let tf = (f12_proc+u12_proc)/10
if (f13_proc+u13_proc)/10 > tf then let tf = (f13_proc+u13_proc)/10
if (f14_proc+u14_proc)/10 > tf then let tf = (f14_proc+u14_proc)/10
if (f15_proc+u15_proc)/10 > tf then let tf = (f15_proc+u15_proc)/10
if (f16_proc+u16_proc)/10 > tf then let tf = (f16_proc+u16_proc)/10

let tf = tf*0.8
skip 3 lines
let x = 150000
if tf>x then let x = 300000
if tf>x then let x = 750000
if tf>x then let x = 1500000
let xdiv = x/30
for i = 1 to 30 do begin
  if i=1 or i=11 or i=21 then print 4 spaces, x/1000 using "#####";
  if i=15 then print "RDTE ($ 1M)";
  if i=17 then print "PROC ($10M)";
  print column 13, "!";

  if f0_tb + u0_tb >= x then begin
    if f0_tb >= x then print fx; else print ufx; end
    else print " ";
  if f0_dev + u0_dev >= x then begin
    if f0_dev >= x then print fd; else print ufd; end
    else print " ";
  if f1_proc+u1_proc >= x*10 then begin
    if f1_proc >= x*10 then print fp; else print ufp; end
    else print " ";
  print 1 space;

  if f1_tb + u1_tb >= x then begin
    if f1_tb >= x then print fx; else print ufx; end
    else print " ";
  if f1_dev + u1_dev >= x then begin
    if f1_dev >= x then print fd; else print ufd; end
    else print " ";
  if f2_proc+u2_proc >= x*10 then begin
    if f2_proc >= x*10 then print fp; else print ufp; end
    else print " ";
  print 1 space;

  if f2_tb + u2_tb >= x then begin
    if f2_tb >= x then print fx; else print ufx; end
    else print " ";
  if f2_dev + u2_dev >= x then begin
    if f2_dev >= x then print fd; else print ufd; end
    else print " ";
  if f3_proc+u3_proc >= x*10 then begin
    if f3_proc >= x*10 then print fp; else print ufp; end

```

```
    else print " ";
    print 1 space;

    if f3_tb + u3_tb >= x then begin
      if f3_tb >= x then print fx; else print ufx; end
      else print " ";
    if f3_dev + u3_dev >= x then begin
      if f3_dev >= x then print fd; else print ufd; end
      else print " ";
    if f4_proc+u4_proc >= x*10 then begin
      if f4_proc >= x*10 then print fp; else print ufp; end
      else print " ";
    print 1 space;

    if f4_tb + u4_tb >= x then begin
      if f4_tb >= x then print fx; else print ufx; end
      else print " ";
    if f4_dev + u4_dev >= x then begin
      if f4_dev >= x then print fd; else print ufd; end
      else print " ";
    if f5_proc+u5_proc >= x*10 then begin
      if f5_proc >= x*10 then print fp; else print ufp; end
      else print " ";
    print 1 space;

    if f5_tb + u5_tb >= x then begin
      if f5_tb >= x then print fx; else print ufx; end
      else print " ";
    if f5_dev + u5_dev >= x then begin
      if f5_dev >= x then print fd; else print ufd; end
      else print " ";
    if f6_proc+u6_proc >= x*10 then begin
      if f6_proc >= x*10 then print fp; else print ufp; end
      else print " ";
    print 1 space;

    if f6_tb + u6_tb >= x then begin
      if f6_tb >= x then print fx; else print ufx; end
      else print " ";
    if f6_dev + u6_dev >= x then begin
      if f6_dev >= x then print fd; else print ufd; end
      else print " ";
    if f7_proc+u7_proc >= x*10 then begin
      if f7_proc >= x*10 then print fp; else print ufp; end
      else print " ";
    print 1 space;

    if f7_tb + u7_tb >= x then begin
      if f7_tb >= x then print fx; else print ufx; end
      else print " ";
    if f7_dev + u7_dev >= x then begin
      if f7_dev >= x then print fd; else print ufd; end
      else print " ";
    if f8_proc+u8_proc >= x*10 then begin
      if f8_proc >= x*10 then print fp; else print ufp; end
      else print " ";
```

```

print 2 spaces;

if f9_proc+u9_proc >= x*10 then begin
  if f9_proc >= x*10 then print fp; else print ufp; end
  else print " ";
print 2 spaces;
if f10_proc+u10_proc >= x*10 then begin
  if f10_proc >= x*10 then print fp; else print ufp; end
  else print " ";
print 2 spaces;
if f11_proc+u11_proc >= x*10 then begin
  if f11_proc >= x*10 then print fp; else print ufp; end
  else print " ";
print 2 spaces;
if f12_proc+u12_proc >= x*10 then begin
  if f12_proc >= x*10 then print fp; else print ufp; end
  else print " ";
print 2 spaces;
if f13_proc+u13_proc >= x*10 then begin
  if f13_proc >= x*10 then print fp; else print ufp; end
  else print " ";
print 2 spaces;
if f14_proc+u14_proc >= x*10 then begin
  if f14_proc >= x*10 then print fp; else print ufp; end
  else print " ";
print 2 spaces;
if f15_proc+u15_proc >= x*10 then begin
  if f15_proc >= x*10 then print fp; else print ufp; end
  else print " ";
print 2 spaces;
if f16_proc+u16_proc >= x*10 then begin
  if f16_proc >= x*10 then print fp; else print ufp; end
  else print " ";
print " ";
let x = x-xdiv
end
print 13 spaces;
for i = 14 to 132 do print "-";
print " ";
print 16 spaces, "1986", 6 spaces, "1987", 6 spaces, "1988", 6 spaces,
  "1989", 6 spaces, "1990", 6 spaces, "1991", 6 spaces, "1992", 6 spaces, "1993",
  " ; 94 95 96 97 98 99 00 01"
print column 94, "!"
print "Tech Base ", f0_tb using form1, f1_tb using form1, f2_tb using form1,
  f3_tb using form1, f4_tb using form1, f5_tb using form1,
  f6_tb using form1, f7_tb using form1, column 94, "!"
print " (", fx, "/", ufx, " ) ", -u0_tb using form2, -u1_tb using form2,
  -u2_tb using form2, -u3_tb using form2, -u4_tb using form2,
  -u5_tb using form2, -u6_tb using form2, -u7_tb using form2, column 94, "!"
print column 94, "!"
print "Development ", f0_dev using form1, f1_dev using form1,
  f2_dev using form1, f3_dev using form1, f4_dev using form1,
  f5_dev using form1, f6_dev using form1, f7_dev using form1, column 94, "!"
print " (", fd, "/", ufd, " ) ", -u0_dev using form2, -u1_dev using form2,
  -u2_dev using form2, -u3_dev using form2, -u4_dev using form2,
  -u5_dev using form2, -u6_dev using form2, -u7_dev using form2, column 94, "!"

```

```
print column 94, ":"
print "Procurement ", f1_proc using form1, f2_proc using form1,
      f3_proc using form1, f4_proc using form1, f5_proc using form1,
      f6_proc using form1, f7_proc using form1, f8_proc using form1,
      column 94, "!"      See Out Year Procurement"
print column 94, ":"
print 54 spaces;
for i=95 to 132 do print "-";
print ""
print 56 spaces, "1994", 6 spaces, "1995", 6 spaces, "1996", 6 spaces, "1997",
      6 spaces, "1998", 6 spaces, "1999", 6 spaces, "2000", 6 spaces, "2001"
skip 1 line
print 29 spaces, "Procurement (Out Years)", f9_proc using form1,
      f10_proc using form1, f11_proc using form1, f12_proc using form1,
      f13_proc using form1, f14_proc using form1, f15_proc using form1,
      f16_proc using form1
print 52 spaces, -u9_proc using form1, -u10_proc using form1,
      -u11_proc using form1, -u12_proc using form1, -u13_proc using form1,
      -u14_proc using form1, -u15_proc using form1, -u16_proc using form1

end
```

Jul 25 12:16 1986 defroll Page 1

{Deficiency Summary

Last changed 2/4/86. added report mission area, system tradoc mission area,
and date-time stamp.
2/7/86. added page letter parameter. moved page number to input. changes to
ssndesc .
5/20/86. changed to mamp database.
5/29/86. changed to include all systems.
6/10/86 revised the read statements.
6/13/86 added def_class for classification. corrected printing of cont_value.
7/23/86 changed the workpackage input variables to generic years.
7/25/86. changed to generic rdte rollup file fundpro (from syspro1). }

database mamp end

```
define
  variable linesleft      type integer
  variable counter        type integer
  variable total_fund     type integer
  variable pagebreak      type integer
  variable pagetrail      type character length 1
  variable stars          type character length 4
  variable evalcount      type integer
  variable firstfl       type integer
  variable pno            type integer
  variable break          type integer
  variable obreak        type integer
  param[1] pletter       type character length 1
  param[2] code           type integer
  param[3] name          type character length 12
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  left margin 0
  right margin 132
  report to "defroll.out"
end

read into b
  dcl_def
  where dcl_code = code and dcl_name = name
end

read into a
  b
  def_desc def_type def_class
  ssndef_ssn ssndef_cont_value
  joining b.dcl_def = optional def_def
  and b.dcl_def = optional ssndef_def
end

read into c
```



```

a
prior1_ndef
ssn_ssn ssn_acq_code ssn_amc_mgr ssn_da_ma ssn_tradoc_ma
lrrdp_title lrrdp_procfl
lrrdp_procf2 lrrdp_procf3 lrrdp_procf4 lrrdp_procf5 lrrdp_procf6
lrrdp_procf7 lrrdp_procf8 lrrdp_procf9 lrrdp_procf10 lrrdp_procf11
lrrdp_procf12 lrrdp_procf13 lrrdp_procf14 lrrdp_procf15 lrrdp_procf16
f0_tb fl_tb f2_tb f3_tb f4_tb f5_tb f6_tb f7_tb
f0_dev fl_dev f2_dev f3_dev f4_dev f5_dev f6_dev f7_dev
joining a. ssndef_ssn = optional prior1_ssn_no
        and a. ssndef_ssn = optional ssn_ssn
        and a. ssndef_ssn = optional lrrdp_ssn
        and a. ssndef_ssn = optional fp_ssn
end

sort by dcl_def ssn_acq_code ssndef_cont_value ssndef_ssn end

format

page header
let pagetrail=def_class
if pagetrail="S" then print column 54, "***** S E C R E T *****"
else print column 47, "***** C O N F I D E N T I A L *****"
if pagebreak=1 then begin
    skip 1 line
    print "TRADOC DEFICIENCY:", dcl_def;
    print "      (continued)";
    print column 50, name, 2 spaces;
    if def_type = 1 then print "Primary"
    else if def_type = 2 then print "Related"
    else if def_type = 3 then print "Non-Materiel"
    else if def_type = 4 then print "Health Service"
    else print " "
    skip 1 line
    print column 61, "-----",
    "-----"
    print column 61, "! CON ! 86 87 88 89 90 91 92 93 94 95 96 ",
    "97 98 99 00 01 !"
    print column 61, "!-----",
    "-----!"
end
else begin
    skip 4 lines
    print "TRADOC DEFICIENCY:", dcl_def;
    print column 50, name, 2 spaces;
    if def_type = 1 then print "Primary"
    else if def_type = 2 then print "Related"
    else if def_type = 3 then print "Non-Materiel"
    else if def_type = 4 then print "Health Service"
    else print " "
    skip 1 line
end
let linesleft = 48

before group of dcl_def
let pagebreak = 0

```

```

let firstfl = 0
skip to top of page
print "DESCRIPTION: ";
let break = 118
while def_desc[break, break] <> " " do let break = break-1
print def_desc[1, break]
let obreak = break+1
let break = break+125
while def_desc[break, break] <> " " do let break = break-1
print 7 spaces, def_desc[obreak, break]
let obreak = break+1
let break = break+125
while def_desc[break, break] <> " " do let break = break-1
print 7 spaces, def_desc[obreak, break]
let obreak = break+1
let break = break+125
while def_desc[break, break] <> " " do let break = break-1
print 7 spaces, def_desc[obreak, break]
let obreak = break+1
let break = break+125
while def_desc[break, break] <> " " do let break = break-1
print 7 spaces, def_desc[obreak, break]
skip 2 lines
print column 61, "-----",
"-----"
print column 61, " : CON : 86 87 88 89 90 91 92 93 94 95 96 ",
"97 98 99 00 01 !"
print column 61, " :-----",
"-----!"
let linesleft = linesleft-10

before group of ssn_acq_code
if linesleft < 8 then begin
let pagebreak = 1
if linesleft > 1 then print column 61, ":", column 67, ":", column 132, "!"
print column 61, "-----",
"-----";
skip to top of page
end
if firstfl = 1 then begin
print column 61, ":", column 67, ":", column 132, "!"
print column 61, ":", column 67, ":", column 132, "!"
end
else let firstfl = 1
let counter=0
if ssn_acq_code = 1 then begin
print "Base Case Systems:", column 61, ":", column 67, ":", column 132, "!"
print "-----"; end
else if ssn_acq_code = 2 then begin
print "Type Classified:", column 61, ":", column 67, ":", column 132, "!"
print "-----"; end
else if ssn_acq_code = 3 then begin
print "Development Systems:", column 61, ":", column 67, ":", column 132, "!"
print "-----"; end
else if ssn_acq_code = 4 then begin
print "PIPs:", column 61, ":", column 67, ":", column 132, "!"

```

```

print "-----"; end
else if ssn_acq_code = 5 then begin
print "Technology Demonstrators:", column 61, "!", column 67, "!", column 132, "!"
print "-----"; end
else if ssn_acq_code = 6 then begin
print "Broad Base Tech Area:", column 61, "!", column 67, "!", column 132, "!"
print "-----"; end
else if ssn_acq_code = 7 then begin
print "Requirement Above Corps", column 61, "!", column 67, "!", column 132, "!"
print "-----";
end
print column 61, "!", column 67, "!", column 132, "!"
let linesleft = linesleft-4

before group of ssndef_ssn
if linesleft<4 then begin
let pagebreak=1
if linesleft > 1 then print column 61, "!", column 67, "!", column 132, "!"
print column 61, " -----",
"----- ";
skip to top of page
end
let counter=counter+1
print column 61, "!", column 67, "!", column 132, "!"
print counter, " ", lrrdp_title[1, 50], column 61, "!", " ",
ssndef_cont_value, column 67, "!"
let total_fund=lrrdp_proc1 + lrrdp_proc2+lrrdp_proc3 + lrrdp_proc4+
lrrdp_proc5 + lrrdp_proc6+lrrdp_proc7 + lrrdp_proc8+
lrrdp_proc9 + lrrdp_proc10+lrrdp_proc11 + lrrdp_proc12+
lrrdp_proc13 + lrrdp_proc14+lrrdp_proc15 + lrrdp_proc16
if ssndef_ssn matches "4*" then print column 93, "<STOCK FUNDED>";
else if total_fund = 0 then print column 93, "<NOT SCHEDULED>";
else begin
if prior1_ndef=1 then let stars="****" else let stars="mmmm"
if lrrdp_proc1 > 0 then print stars; else print " ";
if lrrdp_proc2 > 0 then print stars; else print " ";
if lrrdp_proc3 > 0 then print stars; else print " ";
if lrrdp_proc4 > 0 then print stars; else print " ";
if lrrdp_proc5 > 0 then print stars; else print " ";
if lrrdp_proc6 > 0 then print stars; else print " ";
if lrrdp_proc7 > 0 then print stars; else print " ";
if lrrdp_proc8 > 0 then print stars; else print " ";
if lrrdp_proc9 > 0 then print stars; else print " ";
if lrrdp_proc10 > 0 then print stars; else print " ";
if lrrdp_proc11 > 0 then print stars; else print " ";
if lrrdp_proc12 > 0 then print stars; else print " ";
if lrrdp_proc13 > 0 then print stars; else print " ";
if lrrdp_proc14 > 0 then print stars; else print " ";
if lrrdp_proc15 > 0 then print stars; else print " ";
if lrrdp_proc16 > 0 then print stars; else print " ";
end
print column 132, "!"
print column 11, "ssn number: ", ssndef_ssn, column 30, "command: ",
ssn_amc_mgr[1, 9], " - ", ssn_da_ma, " / ", ssn_tradoc_ma,
column 61, "!", column 67, "!"
let linesleft = linesleft-3

```

```
after group of ssndef_ssn
  let evalcount = 0
  if f0_tb+f0_dev > 0 then print"++++"; else print " ";
  if f1_tb+f1_dev > 0 then print"++++"; else print " ";
  if f2_tb+f2_dev > 0 then print"++++"; else print " ";
  if f3_tb+f3_dev > 0 then print"++++"; else print " ";
  if f4_tb+f4_dev > 0 then print"++++"; else print " ";
  if f5_tb+f5_dev > 0 then print"++++"; else print " ";
  if f6_tb+f6_dev > 0 then print"++++"; else print " ";
  if f7_tb+f7_dev > 0 then print"++++"; else print " ";
  print column 132, ":"

after group of dcl_def
  let pagebreak = 0
  if linesleft>1 then print column 61, ":", column 67, ":", column 132, ":"
  print column 61, "-----",
    "-----";

page trailer
  skip 1 line
  print "Legend:";
  if pagetrail = "S" then print column 54, "***** S E C R E T *****"
  else print column 47, "***** C O N F I D E N T I A L *****"
  print " *** - System Production Funding", column 100, date, 2 spaces, time
  print " mmm - Multi-def. System Production Funding"
  print " +++ - Work Package RDTE Funding";
  if pletter="Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1

end
```

Jun 30 16:09 1986 defsys Page 1

(deficiency to system cross reference check. 6/5/86
6/30/86 changed to unclassified with new guidance.)

database mamp end

define

variable pno type integer
param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12

end

input

prompt for pno using "Please enter the starting page number > "
end

output

left margin 0
right margin 132
report to "defsys.out"
end

read into a

dcl_def
where dcl_code = code and dcl_name = name
end

read into b

a
ssndef_ssn ssndef_cont_value
lrrdp_title
joining a.dcl_def = optional ssndef_def
and ssndef_ssn = optional lrrdp_ssn
end

read into c

scl_ssn
where scl_code = code and scl_name = name
end

read into d

b
c
joining b.ssn = optional c.scl_ssn
end

sort by dcl_def ssndef_ssn end

format

page header

print column 47, "***** U N C L A S S I F I E D *****"
skip 2 lines
print column 44, name clipped, " DEFICIENCIES WITH SYSTEM BDP RANKING"
skip 2 lines

```
page trailer
  skip 2 lines
  print column 47, "***** U N C L A S S I F I E D *****"
  print column 100, date, 2 spaces, time
  if pletter="Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1

before group of dcl_def
  print dcl_def

before group of ssndef_ssn
  print 10 spaces, ssndef_ssn, 1 space;
  if scl_ssn<>" " then print "*"; else print " ";
  print 5 spaces, ssndef_cont_value, 5 spaces, lrrdp_title

end
```

Jul 29 12:16 1986 fundsysbdp Page 1

(Funded workpackage data for systems in BDP priority order. This file was taken from decsysbdp on 1 May 86. It was requested by Dr. Ken Oscar to support the Belvoir review process. It includes only the funded workpkgs from the year 87 to 92.)

database mamp end

define

```
variable pno      type integer
variable cnt      type integer
variable tot87    type long
variable tot88    type long
variable tot89    type long
variable tot90    type long
variable tot91    type long
variable tot92    type long
param[1] pletter type character length 1
param[2] code     type integer
param[3] name     type character length 12
```

end

input

```
prompt for pno using "Please enter the starting page number > "
end
```

output

```
left margin 0
right margin 132
report to "fundsysbdp.out"
end
```

read into b

```
scl_ssn
where scl_code = code and scl_name = name
end
```

read into a

```
b
ssn_anc_mgr ssn_da_ma ssn_tradoc_ma
lrrdp_title priori_score priori_ndef
where ssn_acq_code > 2 or ssn_acq_code = 0
joining b.scl_ssn = optional ssn_ssn
and b.scl_ssn = optional lrrdp_ssn
and b.scl_ssn = optional priori_ssn_no
end
```

read into c

```
b
wkpkg_wkpsys_idx wkpkg_title wkpf_yr wkpf_fund
where wkpf_fund > 0 and wkpf_yr > 1986
joining b.scl_ssn = wkpsys_ssn
and wkpsys_wkpkg_idx = wkpkg_wkpsys_idx
and wkpkg_no_idx = wkpf_wkpkg_idx
end
```

read into d

```
a
c.wkpkg_wkpsys_idx c.wkpkg_title c.wkpf_yr c.wkpf_fund
joining a.scl_ssn = optional c.scl_ssn
end

sort by priori_score descending priori_ndef descending
scl_ssn wkpkg_no wkpf_yr end

format

page header
print column 47, "***** U N C L A S S I F I E D *****"
skip 2 lines
print column 44, name clipped, " I-N DEVELOPMENT SYSTEM BDP RANKING"
print column 43, "WITH FUNDED RDT&E WORKPACKAGE TOTALS SHOWN"
skip 2 lines
print 6 spaces, "SSN", 3 spaces, "DA MA", 2 spaces, "AMC MGR", 15 spaces,
"TITLE", column 86, "FUNDED DEVELOPMENT SCHEDULE ($K)", column 127, "RATING"
print column 78, "FY87 FY88 FY89 FY90 FY91 FY92"
skip 1 line

page trailer
skip 2 lines
print column 47, "***** U N C L A S S I F I E D *****"
print column 100, date, 2 spaces, time
if pletter="Z" then print column 60, pno
else print column 60, pletter, "-", pno using "###"
let pno = pno+1

before group of scl_ssn
let cnt = cnt+1
let tot87 = 0
let tot88 = 0
let tot89 = 0
let tot90 = 0
let tot91 = 0
let tot92 = 0
print cnt using "###. ", scl_ssn, 2 spaces, ssn_da_ma,
3 spaces, ssn_amc_mgr[1,8], 1 space, lrrdp_title[1,44], 2 spaces;

after group of scl_ssn
print tot87 using "***** ", tot88 using "***** ", tot89 using "***** ",
tot90 using "***** ", tot91 using "***** ", tot92 using "***** ",
priori_score using "*****.##"

before group of wkpf_yr
if wkpf_yr = 1987 then let tot87 = tot87 + wkpf_fund
else if wkpf_yr = 1988 then let tot88 = tot88 + wkpf_fund
else if wkpf_yr = 1989 then let tot89 = tot89 + wkpf_fund
else if wkpf_yr = 1990 then let tot90 = tot90 + wkpf_fund
else if wkpf_yr = 1991 then let tot91 = tot91 + wkpf_fund
else if wkpf_yr = 1992 then let tot92 = tot92 + wkpf_fund

end
```


(DECISION AID FOR WORKPACKAGES BASED ON BDP PRIORITIES.
THIS REPORT SHOWS THE FUNDED WORKPACKAGES.

4/25/86 changed to mamp data base.
taken from decwpbdp, altered to show the workpackage funded
quantities, and sorted by the subcategory.
5/12/86 limited to 6.3, 6.4 and 6.7 workpackages.)

database mamp end

define

variable pno type integer
variable pgend type integer
variable cnt type integer
param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12

end

input

prompt for pno using "Please enter the starting page number > "
end

output

left margin 0
right margin 132
report to "fundwpbdp.out"

end

read into b

pcl_idx
where pcl_code = code and pcl_name = name
and (pcl_subcat = "6.3A" or pcl_subcat = "6.3B" or pcl_subcat = "6.4"
or pcl_subcat = "6.7")

end

read into c

prior2_score prior2_nsys
wkpkg_no_idx wkpkg_subcat wkpkg_pe wkpkg_proj wkpkg_task wkpkg_title
joining b.pcl_idx = wkpkg_proj_idx
and wkpkg_no_idx = optional prior2_wkpkg_idx

end

read into d

wkpf_wkpkg_idx wkpf_yr wkpf_fund
where wkpf_fund > 0
joining c.wkpkg_no_idx = wkpf_wkpkg_idx

end

read into a

c
d.wkpf_yr d.wkpf_fund
joining c.wkpkg_no_idx = optional d.wkpf_wkpkg_idx

end

sort by wkpkg_subcat prior2_score descending prior2_nsys descending

```
      wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no wkpf_yr end

format

page header
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 47, name clipped, " I T O N ", wkpkg_subcat clipped,
    " WORKPACKAGE BDP RANKINGS"
  print column 50, "WITH FUNDED FISCAL RESOURCES SHOWN"
  skip 3 lines
  print column 87, "FUNDED DEVELOPMENT SCHEDULE"
  print "-----"
  print "
  print ":", PE/PROJ/TASK/WKPKG, 5 spaces, "TITLE", column 67,
    ":", FY85   FY86   FY87   FY88   FY89   FY90   FY91   FY92 ":",
  print "-----"
  print ":", column 67, ":", column 132, ":",
  let pgend = 0

page trailer
  if pgend = 0 then begin
    print ":", column 67, ":", column 132, ":",
    print "-----"
  end
  else skip 2 lines
  skip 2 lines
  print column 47, "***** U N C L A S S I F I E D *****"
  print column 100, date, 2 spaces, time
  if pletter = "Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1

before group of wkpkg_subcat
  let cnt = 0
  skip to top of page

after group of wkpkg_subcat
  let pgend = 1
  print ":", column 67, ":", column 132, ":",
  print "-----"

before group of wkpkg_no
  let cnt = cnt+1
  print ":", cnt using "###. ", wkpkg_title[1,55], column 67, ":",

before group of wkpf_yr
  if wkpf_yr > 0 then begin
    if wkpf_fund > 0 then print column (8*(wkpf_yr-1985)+68),
      wkpf_fund using "##### ";
  end

after group of wkpkg_no
```

Jun 3 14:32 1986 fundwpsdp Page 3

```
print column 132, "!"  
print "!", 5 spaces, wkpkg_cmd, 1 space,  
      wkpkg_pe, 1 space, wkpkg_proj, 1 space, wkpkg_task clipped, 1 space,  
      wkpkg_no, column 67, "!", column 132, "!"  
end
```

(DECISION AID FOR WORKPACKAGES BASED ON BDP PRIORITIES.
THIS REPORT SHOWS THE FUNDED WORKPACKAGES.

4/25/86 changed to mamp data base.
taken from decwpbdp, altered to show the workpackage funded
quantities, and sorted by the subcategory.
5/12/86 taken from fundwpbdp.
changed to include only 6.3 and 6.4 and 6.7 workpackages, eliminate
subcategory sort, added extra spacing after each workpackage.)

database mamp end

```
define
  variable pno      type integer
  variable pgen     type integer
  variable cnt      type integer
  param[1] pletter  type character length 1
  param[2] code     type integer
  param[3] name     type character length 12
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  left margin 0
  right margin 132
  report to "fundwpbdp2.out"
end

read into b
  pcl_idx
  where pcl_code = code and pcl_name = name
     and (pcl_subcat = "6.3A" or pcl_subcat = "6.3B" or pcl_subcat = "6.4"
        or pcl_subcat = "6.7")
end

read into c
  prior2_score prior2_nsys
  wkpkg_no_idx wkpkg_pe wkpkg_proj wkpkg_task wkpkg_title
  joining b.pcl_idx = wkpkg_proj_idx
     and wkpkg_no_idx = optional prior2_wkpkg_idx
end

read into d
  wkpf_wkpkg_idx wkpf_yr wkpf_fund
  where wkpf_fund > 0
  joining c.wkpkg_no_idx = wkpf_wkpkg_idx
end

read into a
  c
  d.wkpf_yr d.wkpf_fund
  joining c.wkpkg_no_idx = optional d.wkpf_wkpkg_idx
end
```

```
sort by prior2_score descending prior2_nsys descending
      wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no wkpf_yr end

format

page header
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 49, name clipped, " 1 TO N WORKPACKAGE BDP RANKINGS"

  print column 50, "WITH FUNDED FISCAL RESOURCES SHOWN"
  skip 2 lines
  print column 87, "FUNDED DEVELOPMENT SCHEDULE"
  print "-----"
  print "
  |          PE/PROJ/TASK/WKPKG", 5 spaces, "TITLE", column 67,
  |  FY85   FY86   FY87   FY88   FY89   FY90   FY91   FY92  |"
  print "-----"
  print "
  |-----|"
  print "!", column 67, "!", column 132, "!"
  let pgend = 0

page trailer
  if pgend = 0 then begin
    print "-----"
    end
  else skip 1 line
  skip 2 lines
  print column 47, "***** U N C L A S S I F I E D *****"
  print column 100, date, 2 spaces, time
  if pletter = "Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1

on last record
  print "-----"
  let pgend = 1

before group of wkpkg_no
  let cnt = cnt+1
  print "!", cnt using "###. ", wkpkg_title[1, 55], column 67, "!"

before group of wkpf_yr
  if wkpf_yr > 0 then begin
    if wkpf_fund > 0 then print column (8*(wkpf_yr-1985)+68),
      wkpf_fund using "***** ";
  end

after group of wkpkg_no
  print column 132, "!"
  print "!", 5 spaces, wkpkg_cmd, 1 space,
    wkpkg_pe, 1 space, wkpkg_proj, 1 space, wkpkg_task clipped, 1 space,
    wkpkg_no, column 67, "!", column 132, "!"
```

Jun 3 14:32 1986 fundwpbdp2 Page 3

```
print ":", column 67, ":", column 132, ":"  
end
```

(DECISION AID FOR WORKPACKAGES BASED ON LRRDP PRIORITIES.
THIS REPORT SHOWS THE FUNDED WORKPACKAGES.

4/25/86 changed to mamp data base.
taken from decupbdb, altered to show the workpackage funded
quantities, and sorted by the subcategory.
5/12/86 taken from fundupbdb and lrrdpwp, removed subcategory sort, limited
to 6.3A to 6.4 and 6.7 workpackages, added report spacing. }

database mamp end

```
define
  variable pno      type integer
  variable pgen     type integer
  variable cnt      type integer
  param[1] pletter  type character length 1
  param[2] code     type integer
  param[3] name     type character length 12
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  left margin 0
  right margin 132
  report to "fundwplrp.out"
end

read into b
  pcl_idx
  where pcl_code = code and pcl_name = name
  and (pcl_subcat = "6.3A" or pcl_subcat = "6.3B" or pcl_subcat = "6.4"
  or pcl_subcat = "6.7")
end

read into a
  wkpkg_no_idx wkpkg_pe wkpkg_proj wkpkg_task wkpkg_title wkpkg_pdip
  joining b.pcl_idx = wkpkg_proj_idx
end

read into f
  a
  lrrdppri_pri
  ok = 1
  joining a.wkpkg_pdip = lrrdppri_pdip
end

read into g
  a
  lrrdppri_pri
  ok = 0
  where lrrdppri_pri = 0.0
  joining a.wkpkg_pdip = optional lrrdppri_pdip
end
```

```
assign h = f union g end

read into d
  wkpf_wkpkg_idx wkpf_yr wkpf_fund
  where wkpf_fund > 0
  joining a.wkpkg_no_idx = wkpf_wkpkg_idx
end

read into e
  h
  d.wkpf_yr d.wkpf_fund
  joining h.wkpkg_no_idx = optional d.wkpf_wkpkg_idx
end

sort by ok descending lrrdppri_pri
  wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no wkpf_yr end

format

page header
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 49, name clipped, " 1 TO N WORKPACKAGE PDIP RANKINGS"
  print column 50, "WITH FUNDED FISCAL RESOURCES SHOWN"
  skip 2 lines
  print column 87, "FUNDED DEVELOPMENT SCHEDULE"
  print "-----",
  "-----"
  print " | PE/PROJ/TASK/WKPKG", 5 spaces, "TITLE", column 67,
  " | FY85 FY86 FY87 FY88 FY89 FY90 FY91 FY92 |",
  print " |-----",
  " |-----"
  print " |", column 67, " |", column 132, " |"
  let pgend = 0

page trailer
  if pgend = 0 then begin
    print "-----",
    "-----"
  end
  else skip 1 line
  skip 2 lines
  print column 47, "***** U N C L A S S I F I E D *****"
  print column 100, date, 2 spaces, time
  if pletter = "Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1

on last record
  print "-----",
  "-----"
  let pgend = 1

before group of wkpkg_no
  let cnt = cnt+1
```



```
print "!" cnt using "###. ", wkpkg_title[1,55], column 67, "!"  
before group of wkpfg_yr  
if wkpfg_yr>0 then begin  
  if wkpfg_fund>0 then print column (8*(wkpfg_yr-1985)+68),  
    wkpfg_fund using "##### "  
end  
after group of wkpfg_no  
print column 132, "!"  
print "!", 5 spaces, wkpkg_cmd, 1 space,  
  wkpkg_pe, 1 space, wkpkg_proj, 1 space, wkpkg_task clipped, 1 space,  
  wkpkg_no, column 67, "!", column 132, "!"  
print "!", column 67, "!", column 132, "!"  
end
```

Jun 3 14:32 1986 hipri Page 1

{ report to generate the highest priority pdip for all systems in the control
file to be input directly into hipri file using load ascii. }

database map end

output
page length 32000
top margin 0
left margin 0
right margin 132
report to "hipri.out"
end

read into c
unissn_ssn
lrrdp_title lrrdp_pdip lrrdppri_pri
ok = 1
joining unissn_ssn = lrrdp_ssn
and lrrdp_pdip = lrrdppri_pdip
end

read into b
unissn_ssn
lrrdp_title lrrdp_pdip lrrdppri_pri
ok = 0
where lrrdppri_pri = 0.0
joining unissn_ssn = optional lrrdp_ssn
and lrrdp_pdip = optional lrrdppri_pdip
end

assign a = c union b end

sort by unissn_ssn ok descending lrrdppri_pri end

format

before group of unissn_ssn
print unissn_ssn, "!", lrrdp_pdip, "!", lrrdppri_pri, "!", ok, "!", lrrdp_title, "!"

end

```
database mamp end

define
  variable cnt type integer
  variable flag type integer
  variable v type character length 1
end

output
  top margin 0
  page length 32000
  left margin 0
  right margin 132
  report to "joinassr.out"
end

read into a
  assn
end

sort by assn_ssn assn_assoc_ssn end

format

before group of assn_ssn
  let cnt = 0
  let flag = 0
  let v = ":"
  print assn_ssn,v;

before group of assn_assoc_ssn
  let cnt = cnt+1
  if cnt>15 then let flag = 1
  if flag = 0 then print assn_assoc_ssn,v;

after group of assn_ssn
  let cnt = 15-cnt
  while cnt>0 do begin
    print " ",v;
    let cnt = cnt-1
  end
  print ""

end
```

Jul 1 14:14 1986 lrrdp Page 1

database mamp end

output
 report to "lrrdp.out"
end

read
 lrrdp_ssn lrrdp_title
 ssn_tradoc_ma
 joining lrrdp_ssn = optional ssn_ssn
end

sort by lrrdp_ssn end

format

page header
 print column 50, "LRRDP SYSTEMS BY SSN"
 skip 2 lines

before group of lrrdp_ssn
 print 32 spaces, lrrdp_ssn, 5 spaces, ssn_tradoc_ma, 5 spaces, lrrdp_title
end

Jul 1 14:17 1986 lrrdp1 Page 1

database mamp end

output
 report to "lrrdp1.out"
end

read into a
 lrrdp_ssn lrrdp_title
 ssn_tradoc_ma
 joining lrrdp_ssn = optional ssn_ssn
end

sort by lrrdp_title lrrdp_ssn end

format

page header
 print column 50, "LRRDP SYSTEMS BY TITLE"
 skip 2 lines

before group of lrrdp_ssn
 print 32 spaces, lrrdp_ssn, 5 spaces, ssn_tradoc_ma, 5 spaces, lrrdp_title
end

Jun 3 14:32 1986 lrrdppri Page 1

```
{ This report produces a single list of all lrrdp pdips and priorities for
  use in interpreting the lrrdp priority outputs. }

database mamp end

define
  variable pno      type integer
  param[1] pletter  type character length 1
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  left margin 0
  right margin 132
  report to "lrrdppri.out"
end

read
  lrrdppri
end

sort by lrrdppri_pdip end

format

page header
  print column 47, "***** C O N F I D E N T I A L *****"
  skip 2 lines
  print column 47, " DA PRIORITY BY PDIP AND INCREMENT"
  skip 2 lines
  print 6 spaces, "PDIP/INCREMENT", 6 spaces, "PRIORITY"
  skip 1 line

page trailer
  skip 2 lines
  print column 47, "***** C O N F I D E N T I A L *****"
  print column 100, date, 2 spaces, time
  if pletter="Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1

before group of lrrdppri_pdip
  print 6 spaces, lrrdppri_pdip[1, 4], "-", lrrdppri_pdip[5, 6], 6 spaces, lrrdppri_pri
end
```

(System priority report based on the LRRDP priorities.
4/23/86 changed to mamp database.
4/25/86 changed to reflect classified nature of the lrrdp priorities.
added parameter to select or deselect the priorities.
5/2/86 changed to print only the highest priority pdip for one system.
5/12/86 added the base case systems and stuck them at the head of the list.)

database mamp end

```
define
  variable cnt      type integer
  variable pno      type integer
  param[1] pletter  type character length 1
  param[2] code     type integer
  param[3] name     type character length 12
  param[4] class    type character length 1
end
```

```
input
  prompt for pno using "Please enter the starting page number > "
end
```

```
output
  left margin 0
  right margin 132
  report to "lrrdpsys.out"
end
```

```
read into b
  scl_ssn
  where scl_code = code and scl_name = name
end
```

```
read into a
  b
  ssn_amc_mgr ssn_da_ma
  bc = 0
  where ssn_acq_code > 2 or ssn_acq_code = 0
  joining b.scl_ssn = optional ssn_ssn
end
```

```
read into d
  b
  ssn_amc_mgr ssn_da_ma
  bc = 1
  where ssn_acq_code = 1
  joining b.scl_ssn = ssn_ssn
end
```

```
assign e = a union d end
```

```
read into c
  e
  hipri_pdip hipri_pri hipri_ok hipri_title
  joining e.scl_ssn = hipri_ssn
end
```

```
sort by bc descending hipri_ok descending hipri_pri scl_ssn hipri_pdip end
```

```
format
```

```
page header
```

```
if class = "U" then print column 47, "***** U N C L A S S I F I E D *****"
else print column 47, "***** C O N F I D E N T I A L *****"
skip 2 lines
print column 46, "DA DEVELOPMENT SYSTEM PRIORITY FOR ", name
skip 2 lines
print 17 spaces, "SSN", 5 spaces, "DA MA", 4 spaces, "AMC MGR", 15 spaces,
"TITLE", column 98, " PDIP PRIORITY"
skip 1 line
```

```
page trailer
```

```
skip 2 lines
if class = "U" then print column 47, "***** U N C L A S S I F I E D *****"
else print column 47, "***** C O N F I D E N T I A L *****"
print column 100, date, 2 spaces, time
if pletter="Z" then print column 60, pno
else print column 60, pletter, "-", pno using "###"
let pno = pno+1
```

```
before group of hipri_pdip
```

```
let cnt = cnt+1
print 10 spaces, cnt using "###", " ", scl_ssn, 4 spaces, ssn_da_ma,
5 spaces, ssn_amc_mgr, column 30, hipri_title[1, 50], 2 spaces,
hipri_pdip[1, 4], "-", hipri_pdip[5, 6], 2 spaces;
if bc=1 then print " base case"
else if class = "C" then begin
if hipri_ok=1 then print hipri_pri else print " not assigned"
end
else begin
if hipri_ok=1 then print "" else print " not assigned"
end
end
```

```
end
```


{ Workpackage priority report based on lrrdp priorities.

4/23/86 changed to the mamp data base.

4/25/86 changed to reflect classification of lrrdp priorities and to
add parameter to deselect printing of classification.

5/12/86 limited to 6.3, 6.4, and 6.7 workpackages.}

database mamp end

```
define
  variable cnt      type integer
  variable pno      type integer
  param[1] pletter  type character length 1
  param[2] code     type integer
  param[3] name     type character length 12
  param[4] class    type character length 1
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  left margin 0
  right margin 132
  report to "lrrdpwp.out"
end

read into b
  pcl_idx
  where pcl_code = code and pcl_name = name
     and (pcl_subcat = "6.3A" or pcl_subcat = "6.3B" or pcl_subcat = "6.4"
        or pcl_subcat = "6.7")
end

read into a
  wkpkg_cmd wkpkg_no wkpkg_pe wkpkg_proj wkpkg_task wkpkg_title wkpkg_pdip
  joining b.pcl_idx = wkpkg_proj_idx
end

read into c
  a
  lrrdppri_pri
  ok=1
  joining a.wkpkg_pdip = lrrdppri_pdip
end

read into b
  a
  lrrdppri_pri
  ok = 0
  where lrrdppri_pri = 0.0
  joining a.wkpkg_pdip = optional lrrdppri_pdip
end

assign a = c union b end
```

Jun 3 14:32 1986 lrrdpwp Page 2

```
sort by ok descending lrrdppri_pri
      wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no end
```

format

```
page header
  if class = "U" then print column 47, "***** UNCLASSIFIED *****"
  else print column 47, "***** CONFIDENTIAL *****"
  skip 2 lines
  print column 47, " DA WORKPACKAGE PRIORITY FOR ", name
  skip 2 lines
  print 6 spaces, "PE    PROJ TASK    NO", 18 spaces, "TITLE", 45 spaces,
    "COMMAND", 6 spaces, "PDIP    PRIORITY"
  skip 1 line
```

page trailer

```
skip 2 lines
  if class = "U" then print column 47, "***** UNCLASSIFIED *****"
  else print column 47, "***** CONFIDENTIAL *****"
  print column 100, date, 2 spaces, time
  if pletter="Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1
```

after group of wkpkg_no

```
let cnt = cnt+1
print cnt using "###. ", wkpkg_pe, 1 space, wkpkg_proj, 1 space, wkpkg_task,
  1 space, wkpkg_no, 3 spaces, wkpkg_title, wkpkg_cmd, wkpkg_pdip[1, 4], "-",
  wkpkg_pdip[5, 6];
  if class = "C" then begin
    if ok=1 then print lrrdppri_pri else print " not assigned"
  end
  else begin
    if ok=1 then print "" else print " not assigned"
  end
end
```

end

Jul 16 15:05 1986 mamppdip Page 1

(MAMP PDIP REPORT. 7/16/86.

This report is an abbreviated lrrdp summary that reflects the data in the mamp. It is intended for high level review of the mission area program.)

database mamp end

define

```
variable cnt      type integer
variable cont     type integer
variable break   type integer
variable obreak  type integer
variable pno     type integer
variable unf     type integer
variable stars   type character length 3
param[1] pletter type character length 1
param[2] code    type integer
param[3] name    type character length 12
```

end

input

```
prompt for pno using "Please enter the starting page number > "
end
```

output

```
left margin 0
right margin 132
report to "mamppdip.out"
end
```

read into c

```
sci_ssn
where sci_name = name and sci_code = code
end
```

read into a

```
c
ssn_amc_mgr ssn_tradoc_ma ssn_da_ma
bctc = 0
where ssn_acq_code < 3
joining c.sci_ssn = optional ssn_ssn
end
```

read into b

```
c
ssn_amc_mgr ssn_tradoc_ma ssn_da_ma
bctc = 1
where ssn_acq_code > 2
joining c.sci_ssn = optional ssn_ssn
end
```

assign c = a union b end

read into b

```
c
prior1_score
lrrdp
```

```
lrrdppri
  joining c.scl_ssn = optional priori_ssn_no
         and c.scl_ssn = lrrdp_ssn
         and lrrdp_pdip = optional lrrdppri_pdip
end

read into d
  pcl_idx
  where pcl_cat = "6.3" and pcl_subcat <> "6.5"
         and pcl_code = code and pcl_name = name
end

read into e
  proj_title
  wkpkg_wkpsys_idx wkpkg_pdip
  wkpfr_f0 wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
  wkpfr_u0 wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
  joining d.pcl_idx = wkpkg_proj_idx
         and wkpkg_no_idx = optional wkpfr_wkpkg_idx
         and d.pcl_idx = proj_idx
end

read into c
  b
  e
  joining b.lrrdp_sys_idx = optional uniwkp_sys_idx
         and uniwkp_wkpkg_idx = optional e.wkpkg_wkpsys_idx
end

sort by lrrdp_pdip[1,4] lrrdp_pdip[5,6] bctc priori_score descending
       scl_ssn wkpkg_pe wkpkg_proj wkpkg_cmd end

format

page header
  let cnt = 0
  print column 47, "***** C O N F I D E N T I A L *****"
  skip 2 lines
  print column 52, "MAMP PDIP SUMMARY FOR ", name clipped
  skip 1 line
  print "PDIP: ", lrrdp_pdip[1,4];
  if cnt = 1 then print " (continued)";
  print column 28, "Title: ", lrrdppri_title, column 65,
    "DA/TRADOC Mission Areas: ", lrrdppri_da_ma, " / ", lrrdppri_tradoc_ma,
    column 110, "Proponent: ", lrrdppri_tradoc_pro
  let cnt = 0
  skip 2 lines
  print column 15, "SYSTEM AND RELATED PROJECT TITLES",
    column 87, "PROCUREMENT/RDTE SCHEDULE"
  print column 75,
    "-----"
  print column 75,
    "; 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00 01: BDP ;"
  print column 75,
    "{-----}"
```

```

page trailer
print "Legend:", column 47, "***** C O N F I D E N T I A L *****"
print "***/^/^/^ - Funded/Unfunded Procurement", column 100, date, 2 spaces, time
print "+!+/--- - Funded/Unfunded RDTE";
if pletter="Z" then print column 60, pno
else print column 60, pletter, "-", pno using "###"
let pno = pno+1

before group of lrrdp_pdip[1,4]
skip to top of page

before group of lrrdp_pdip[5,6]
if cnt > 37 then begin
let cont = 1
skip to top of page
end
print column 75, ":", column 124, ":", column 132, "!"
print column 5, "Increment: ", lrrdp_pdip[1,4], "-", lrrdp_pdip[5,6],
column 30, "Priority: ", lrrdp_pri_pri using "###.##";
print column 75, ":", column 124, ":", column 132, "!"
if lrrdp_pri_pri > 800 0 or lrrdp_pri_pri < 1.0 then let unf=1 else let unf=0
let cnt = cnt+2

after group of lrrdp_pdip[5,6]
if cnt < 43 then print column 75, ":", column 124, ":", column 132, "!"
print column 75, "-----"
let cnt = cnt+2

before group of bctc
if cnt > 39 then begin
let cont = 1
print column 75,
"-----";
skip to top of page
end
print column 75, ":", column 124, ":", column 132, "!"
if bctc = 0 then print column 10, "*** Procurement Systems ***";
else print column 10, "*** Development Systems ***";
print column 75, ":", column 124, ":", column 132, "!"
print column 75, ":", column 124, ":", column 132, "!"
let cnt = cnt+3

before group of scl_ssn
if cnt > 42 then begin
let cont = 1
print column 75,
"-----";
skip to top of page
end
let cnt = cnt+1
print scl_ssn, 2 spaces, lrrdp_title[1,55], 1 space,
ssn_da_ma, " / ", ssn_tradoc_ma, column 75, "!"
if unf=1 then let stars="^" else let stars="*"
if lrrdp_proc1 > 0 then print stars; else print " ";
if lrrdp_proc2 > 0 then print stars; else print " ";
if lrrdp_proc3 > 0 then print stars; else print " ";

```

```

if lrrdp_procf4>0 then print stars; else print " ";
if lrrdp_procf5>0 then print stars; else print " ";
if lrrdp_procf6>0 then print stars; else print " ";
if lrrdp_procf7>0 then print stars; else print " ";
if lrrdp_procf8>0 then print stars; else print " ";
if lrrdp_procf9>0 then print stars; else print " ";
if lrrdp_procf10>0 then print stars; else print " ";
if lrrdp_procf11>0 then print stars; else print " ";
if lrrdp_procf12>0 then print stars; else print " ";
if lrrdp_procf13>0 then print stars; else print " ";
if lrrdp_procf14>0 then print stars; else print " ";
if lrrdp_procf15>0 then print stars; else print " ";
if lrrdp_procf16>0 then print stars; else print " ";
if prior1_score = 0.0 then print column 124, "!", column 132, "!"
else print column 124, "!", prior1_score using "#### ", column 132, "!"

after group of wkpkg_cmd
if wkpkg_cmd<>" " then begin
  if cnt > 42 then begin
    let cnt = 1
    print column 75,
    "-----";
    skip to top of page
  end
  let cnt = cnt+1
  print 8 spaces, wkpkg_pe, 1 space, wkpkg_proj, 1 space, wkpkg_cmd[1, 8],
  1 space, proj_title[1, 45], column 75, "!";
  if group total of wkpfr_f0 > 0 then begin
    if group total of wkpfr_u0 > 0 then print "+-+"; else print "+++"; end
  else begin
    if group total of wkpfr_u0 > 0 then print "---"; else print " "; end
  if group total of wkpfr_f1 > 0 then begin
    if group total of wkpfr_u1 > 0 then print "+-+", else print "+++"; end
  else begin
    if group total of wkpfr_u1 > 0 then print "---"; else print " "; end
  if group total of wkpfr_f2 > 0 then begin
    if group total of wkpfr_u2 > 0 then print "+-+"; else print "+++"; end
  else begin
    if group total of wkpfr_u2 > 0 then print "---"; else print " "; end
  if group total of wkpfr_f3 > 0 then begin
    if group total of wkpfr_u3 > 0 then print "+-+"; else print "+++"; end
  else begin
    if group total of wkpfr_u3 > 0 then print "---"; else print " "; end
  if group total of wkpfr_f4 > 0 then begin
    if group total of wkpfr_u4 > 0 then print "+-+"; else print "+++"; end
  else begin
    if group total of wkpfr_u4 > 0 then print "---"; else print " "; end
  if group total of wkpfr_f5 > 0 then begin
    if group total of wkpfr_u5 > 0 then print "+-+"; else print "+++"; end
  else begin
    if group total of wkpfr_u5 > 0 then print "---"; else print " "; end
  if group total of wkpfr_f6 > 0 then begin
    if group total of wkpfr_u6 > 0 then print "+-+"; else print "+++"; end
  else begin
    if group total of wkpfr_u6 > 0 then print "---"; else print " "; end
  if group total of wkpfr_f7 > 0 then begin

```

Jul 16 15:05 1986 mamppdip Page 5

```
      if group total of wkpr_u7 > 0 then print "+--"; else print "+++"; end
      else begin
        if group total of wkpr_u7 > 0 then print "---"; else print " "; end
        print column 124, "!", column 132, "!"
      end
end
```

(6/17/86. SYSTEM/PDIP/COMMODITY report.
7/10/86. changed to unclassified given new guidance.
7/11/86. added classification parameter.)

database mamp end

define

```
variable cnt      type integer
variable head     type integer
variable cont     type integer
variable break   type integer
variable obreak  type integer
variable pno     type integer
param[1] pletter type character length 1
param[2] code    type integer
param[3] name    type character length 12
param[4] class   type character length 1
```

end

input

```
prompt for pno using "Please enter the starting page number > "
end
```

output

```
left margin 0
right margin 132
report to "pdipcom.out"
end
```

read into b

```
scl_ssn
where scl_name = name and scl_code = code
end
```

read into c

```
b
ssn_amc_mgr ssn_acq_code ssn_com_line ssn_tradoc_ma ssn_da_ma
com_acq_strat acq_type
joining b.scl_ssn = optional ssn_ssn
and ssn_acq_code = optional acq_code
and ssn_com_line = optional com_com_line
end
```

read into a

```
c
priori_score
ssndroll
joining c.scl_ssn = optional priori_ssn_no
and c.scl_ssn = optional ssnd_ssn
end
```

read into b

```
a
hipri_title hipri_pdip hipri_pri hipri_ok
lrrdp_proc1 lrrdp_proc2 lrrdp_proc3 lrrdp_proc4 lrrdp_proc5
lrrdp_proc6 lrrdp_proc7 lrrdp_proc8 lrrdp_proc9 lrrdp_proc10
```



```
lrrdp_procfl1 lrrdp_procfl2 lrrdp_procfl3 lrrdp_procfl4 lrrdp_procfl5
lrrdp_procfl6
joining a.scl_ssn = optional hipri_ssn
      and hipri_sys_idx = optional lrrdp_sys_idx
end

read into d
  pcl_idx
  where pcl_cat = "6.3" and pcl_subcat <> "6.5"
      and pcl_code = code and pcl_name = name
end

read into e
  wkpkg_wkpsys_idx wkpkg_pdip
  wkpfr_f0 wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
  wkpfr_u0 wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
  joining d.pcl_idx = wkpkg_proj_idx
      and wkpkg_no_idx = optional wkpfr_wkpkg_idx
end

read into c
  b
  e
  joining b.scl_ssn = optional uniwkp_ssn
      and uniwkp_wkpkg_idx = optional e.wkpkg_wkpsys_idx
end

sort by ssn_com_line ssn_acq_code prior1_score descending
      hipri_ok descending hipri_pri scl_ssn end

format

page header
  let cnt = 0
  if class = "U" then print column 47,"***** UNCLASSIFIED *****"
  else print column 47,"***** CONFIDENTIAL *****"
  skip 2 lines
  print column 46;
  if ssn_com_line <> " " then print ssn_com_line clipped;
      else print "_____";
  print " PDIP COMMODITY SUMMARY FOR ",name clipped;
  if cont = 1 then print " (continued)" else print ""
  let cont = 0
  skip 2 lines
  let head = 1

page trailer
  print "Legend:";
  if class = "U" then print column 47,"***** UNCLASSIFIED *****"
  else print column 47,"***** CONFIDENTIAL *****"
  print "+++ - Funded Procurement", 5 spaces, "~ - LRRDAP Priority",
      column 100,date,2 spaces,time
  print "+++ - Funded RDTE"
  print "--- - Unfunded RDTE";
  if pletter="Z" then print column 60,pno
  else print column 60,pletter, "-",pno using "###"
```

```
let pno = pno+1

before group of ssn_com_line
skip to top of page
print "Acquisition Strategy: ";
let break = 110
while com_acq_strat[break, break]<>" " do let break = break-1
print com_acq_strat[1, break]
let obreak = break+1
let break = break+125
while com_acq_strat[break, break]<>" " do let break = break-1
print 7 spaces, com_acq_strat[obreak, break]
let obreak = break+1
let break = break+125
while com_acq_strat[break, break]<>" " do let break = break-1
print 7 spaces, com_acq_strat[obreak, break]
let obreak = break+1
let break = break+125
while com_acq_strat[break, break]<>" " do let break = break-1
print 7 spaces, com_acq_strat[obreak, break]
let obreak = break+1
let break = break+125
if break>600 then let break = 600
while com_acq_strat[break, break]<>" " do let break = break-1
print 7 spaces, com_acq_strat[obreak, break]
skip 1 lines
let cnt = cnt+3

after group of ssn_com_line
print column 67,
"-----";

before group of scl_ssn
let cnt = cnt+1
if cnt > 22 then begin
let cont = 1
print column 67,
"-----";

skip to top of page
let cnt = 1
end
if head = 1 then begin
print column 15, "SYSTEM TITLES AND BDP DEFICIENCIES",
column 87, "PROCUREMENT/RDTE SCHEDULE"
print column 67,
"-----"
print column 67,
"! PDIP ! 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00 01!PRI/BDP!"
print column 67,
"!-----!";
let head = 0
end
print scl_ssn, 2 spaces, hipri_title[1, 45], 1 space,
ssn_da_ma, " / ", ssn_tradoc_ma, column 67, "!", hipri_pdip[1, 4], "-",
hipri_pdip[5, 6], "!";
if lrrdp_procfl>0 then print "***"; else print " ";
```

```

if lrrdp_proc2>0 then print "***"; else print " ";
if lrrdp_proc3>0 then print "***"; else print " ";
if lrrdp_proc4>0 then print "***"; else print " ";
if lrrdp_proc5>0 then print "***"; else print " ";
if lrrdp_proc6>0 then print "***"; else print " ";
if lrrdp_proc7>0 then print "***"; else print " ";
if lrrdp_proc8>0 then print "***"; else print " ";
if lrrdp_proc9>0 then print "***"; else print " ";
if lrrdp_proc10>0 then print "***"; else print " ";
if lrrdp_proc11>0 then print "***"; else print " ";
if lrrdp_proc12>0 then print "***"; else print " ";
if lrrdp_proc13>0 then print "***"; else print " ";
if lrrdp_proc14>0 then print "***"; else print " ";
if lrrdp_proc15>0 then print "***"; else print " ";
if lrrdp_proc16>0 then print "***"; else print " ";
if ssn_acq_code < 3 then begin
  if class = "C" then
    print column 124, "!", hipri_pri using "#### ", column 132, "!"
  else print column 124, "!", column 132, "!"
end
else print column 124, "!", prior1_score using "#### ", column 132, "!"
print 8 spaces, ssn_amc_mgr[1,7];
if ssnd_def1>0 then begin
  print ssnd_def1 using "####";
  if ssnd_con1<>" " then print "-", ssnd_con1; else print " "; end
if ssnd_def2>0 then begin
  print ssnd_def2 using "####";
  if ssnd_con2<>" " then print "-", ssnd_con2; else print " "; end
if ssnd_def3>0 then begin
  print ssnd_def3 using "####";
  if ssnd_con3<>" " then print "-", ssnd_con3; else print " "; end
if ssnd_def4>0 then begin
  print ssnd_def4 using "####";
  if ssnd_con4<>" " then print "-", ssnd_con4; else print " "; end
if ssnd_def5>0 then begin
  print ssnd_def5 using "####";
  if ssnd_con5<>" " then print "-", ssnd_con5; else print " "; end
if ssnd_def6>0 then begin
  print ssnd_def6 using "####";
  if ssnd_con6<>" " then print "-", ssnd_con6; else print " "; end
if ssnd_def7>0 then begin
  print ssnd_def7 using "####";
  if ssnd_con7<>" " then print "-", ssnd_con7; else print " "; end
print column 67, "!", column 75, "!"
after group of scl_ssn
if group total of wkpfr_f0 > 0 then begin
  if group total of wkpfr_u0 > 0 then print "+-+"; else print "+++"; end
else begin
  if group total of wkpfr_u0 > 0 then print "---"; else print " "; end
if group total of wkpfr_f1 > 0 then begin
  if group total of wkpfr_u1 > 0 then print "+-+"; else print "+++"; end
else begin
  if group total of wkpfr_u1 > 0 then print "---"; else print " "; end
if group total of wkpfr_f2 > 0 then begin
  if group total of wkpfr_u2 > 0 then print "+-+"; else print "+++"; end

```

```
else begin
  if group total of wkpfr_u2 > 0 then print "---"; else print " "; end
  if group total of wkpfr_f3 > 0 then begin
    if group total of wkpfr_u3 > 0 then print "+-+"; else print "+++"; end
  else begin
    if group total of wkpfr_u3 > 0 then print "---"; else print " "; end
    if group total of wkpfr_f4 > 0 then begin
      if group total of wkpfr_u4 > 0 then print "+-+"; else print "+++"; end
    else begin
      if group total of wkpfr_u4 > 0 then print "---"; else print " "; end
      if group total of wkpfr_f5 > 0 then begin
        if group total of wkpfr_u5 > 0 then print "+-+"; else print "+++"; end
      else begin
        if group total of wkpfr_u5 > 0 then print "---"; else print " "; end
        if group total of wkpfr_f6 > 0 then begin
          if group total of wkpfr_u6 > 0 then print "+-+"; else print "+++"; end
        else begin
          if group total of wkpfr_u6 > 0 then print "---"; else print " "; end
          if group total of wkpfr_f7 > 0 then begin
            if group total of wkpfr_u7 > 0 then print "+-+"; else print "+++"; end
          else begin
            if group total of wkpfr_u7 > 0 then print "---"; else print " "; end
            print column 124,"!",column 132,"!"
          end
        end
      end
    end
  end
  before group of ssn_acq_code
  if cnt > 19 then begin
    let cnt = 1
    print column 67,
    "-----";
    skip to top of page
  end
  if head = 1 then begin
    print column 15,"SYSTEM TITLES AND BDP DEFICIENCIES",
    column 87,"PROCUREMENT/RDTE SCHEDULE"
    print column 67,
    "-----"
    print column 67,
    "! PDIP ! 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00 01!PRI/BDP!"
    print column 67,
    "!---!-----!"
    let head = 0
  end
  print column 67,"!",column 75,"!",column 124,"!",column 132,"!"
  print 9 spaces,"***** ",acq_type clipped," *****",
  column 67,"!",column 75,"!",column 124,"!",column 132,"!"
  print column 67,"!",column 75,"!",column 124,"!",column 132,"!"
  let cnt = cnt+2
  after group of ssn_acq_code
  print column 67,"!",column 75,"!",column 124,"!",column 132,"!"
end
```

(6/17/86. PDIP report. Links unique workpackages to systems that are sorted by bdp priority.
7/10/86. changed name to pdipcomda.
7/11/86. added classification parameter.
7/15/86. changed report header to new name.)

database mamp end

define

variable lcnt type integer
variable cnt type integer
variable head type integer
variable cont type integer
variable break type integer
variable obreak type integer
variable pno type integer
variable tot type float

variable pf1 type long
variable pf2 type long
variable pf3 type long
variable pf4 type long
variable pf5 type long
variable pf6 type long
variable pf7 type long

variable rf1 type long
variable rf2 type long
variable rf3 type long
variable rf4 type long
variable rf5 type long
variable rf6 type long
variable rf7 type long

param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12
param[4] class type character length 1

end

input

prompt for pno using "Please enter the starting page number > "
end

output

left margin 0
right margin 132
report to "pdipcomda.out"
end

read into b

sci_ssn
where sci_name = name and sci_code = code
end

read into a

```
      b
      ssn_anc_mgr ssn_acq_code ssn_com_line ssn_tradoc_ma ssn_da_ma
      com_acq_strat
      bctc = 1
      where ssn_acq_code < 3
      joining b.scl_ssn = optional ssn_ssn
             and ssn_com_line = optional com_com_line
    end
```

```
  read into d
    b
    ssn_anc_mgr ssn_acq_code ssn_com_line ssn_tradoc_ma ssn_da_ma
    com_acq_strat
    bctc = 0
    where ssn_acq_code > 2
    joining b.scl_ssn = optional ssn_ssn
           and ssn_com_line = optional com_com_line
  end
```

```
  assign c = a union d end
```

```
  read into a
    c
    ssndroll
    priori_score
    joining c.scl_ssn = optional priori_ssn_no
           and c.scl_ssn = optional ssnd_ssn
  end
```

```
  read into b
    a
    hipri_title hipri_pdip hipri_pri hipri_ok
    lrrdp_procf2 lrrdp_procf3 lrrdp_procf4 lrrdp_procf5
    lrrdp_procf6 lrrdp_procf7 lrrdp_procf8
    joining a.scl_ssn = optional hipri_ssn
           and hipri_sys_idx = optional lrrdp_sys_idx
  end
```

```
  read into d
    pcl_idx
    where pcl_code = code and pcl_name = name and pcl_cat = "6.3"
           and pcl_subcat <> "6.3"
  end
```

```
  read into e
    d
    wkpkg_wkpsys_idx wkpkg_pdip wkpkg_title
    wkpfr_f0 wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
    wkpfr_u0 wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
    joining d.pcl_idx = wkpkg_proj_idx
           and wkpkg_no_idx = optional wkpfr_wkpkg_idx
  end
```

```
  read into c
    b
    e
```

```
uniwkp_srf
joining b.scl_ssn = optional uniwkp_ssn
      and uniwkp_wkpkg_idx = optional e.wkpkg_wkpsys_idx
end

sort by ssn_com_line
      bctc descending
      priori_score descending
      hipri_ok descending
      hipri_pri
      scl_ssn
      wkpkg_proj
      wkpkg_no
end

format

page header
if class = "U" then print column 47,"***** UNCLASSIFIED *****"
else print column 47,"***** CONFIDENTIAL *****"
skip 2 lines
print column 44;
if ssn_com_line <> " " then print ssn_com_line clipped;
                        else print "_____";
print " PDIP COMMODITY DECISION ATR FOR ",name clipped
print column 48,"WITH CUMULATIVE TOTALS (RTDE/PROC)";
if cont = 1 then print " (continued)" else print ""
let cont = 0
skip 2 lines
let head = 1
let lcnt = 0

page trailer
print "Legend:";
if class = "U" then print column 47,"***** UNCLASSIFIED *****"
else print column 47,"***** CONFIDENTIAL *****"
print "~ - LRRDAP Priority",column 100,date,2 spaces,time
print "+ - BDP Rating Score";
if pletter="Z" then print column 60,pno
else print column 60,pletter,"-",pno using "###"
let pno = pno+1

before group of ssn_com_line
let pf1 = 0
let pf2 = 0
let pf3 = 0
let pf4 = 0
let pf5 = 0
let pf6 = 0
let pf7 = 0
let rf1 = 0
let rf2 = 0
let rf3 = 0
let rf4 = 0
let rf5 = 0
let rf6 = 0
```

```

let rf7 = 0
skip to top of page
print "Acquisition Strategy: ";
let break = 110
while com_acq_strat[break, break] <> " " do let break = break-1
print com_acq_strat[1, break]
let obreak = break+1
let break = break+125
while com_acq_strat[break, break] <> " " do let break = break-1
print 7 spaces, com_acq_strat[obreak, break]
let obreak = break+1
let break = break+125
while com_acq_strat[break, break] <> " " do let break = break-1
print 7 spaces, com_acq_strat[obreak, break]
let obreak = break+1
let break = break+125
while com_acq_strat[break, break] <> " " do let break = break-1
print 7 spaces, com_acq_strat[obreak, break]
let obreak = break+1
let break = break+125
if break > 600 then let break = 600
while com_acq_strat[break, break] <> " " do let break = break-1
print 7 spaces, com_acq_strat[obreak, break]
skip 1 lines
let lcnt = lcnt + 7

after group of ssn_com_line
print "!", column 9, "!", column 71, "!", column 79, "!", column 132, "!"
print "-----";
"-----";

before group of scl_ssn
let cnt = 0
if lcnt > 42 then begin
let cont = 1
print "!", column 9, "!", column 71, "!", column 79, "!", column 132, "!"
print "-----";
"-----";

skip to top of page
end
if head = 1 then begin
print "-----";
"-----";

print "!", SSN "!", 6 spaces, "TITLE", column 58, "DA/TDC MA",
column 71, "!", PDIP "!",
" 87 88 89 90 91 92 93 !"
print "!"-----"!";
"-----!";

let lcnt = lcnt + 3
let head = 0
end
print "!", column 9, "!", column 71, "!", column 79, "!", column 132, "!"
print "!", scl_ssn, column 9, "!", 1 space, hipri_title[1, 40], 6 space,
ssn_da_ma, " / ", ssn_tradoc_ma, column 71, "!", hipri_pdip[1, 4], "-",
hipri_pdip[5, 6], "!", column 81;
let pfl = pfl + lrrdp_procfl2

```



```

let pf2 = pf2 + lrrdp_proc3
let pf3 = pf3 + lrrdp_proc4
let pf4 = pf4 + lrrdp_proc5
let pf5 = pf5 + lrrdp_proc6
let pf6 = pf6 + lrrdp_proc7
let pf7 = pf7 + lrrdp_proc8
let tot = lrrdp_proc2 + lrrdp_proc3 + lrrdp_proc4 + lrrdp_proc5 +
          lrrdp_proc6 + lrrdp_proc7 + lrrdp_proc8
if tot = 0 then print column 90. "(Procurement Funds Not Scheduled)";
else begin
  if lrrdp_proc2>0 then print lrrdp_proc2 using "##### ";
  else print " ";
  if lrrdp_proc3>0 then print lrrdp_proc3 using "##### ";
  else print " ";
  if lrrdp_proc4>0 then print lrrdp_proc4 using "##### ";
  else print " ";
  if lrrdp_proc5>0 then print lrrdp_proc5 using "##### ";
  else print " ";
  if lrrdp_proc6>0 then print lrrdp_proc6 using "##### ";
  else print " ";
  if lrrdp_proc7>0 then print lrrdp_proc7 using "##### ";
  else print " ";
  if lrrdp_proc8>0 then print lrrdp_proc8 using "##### ";
  else print " ";
end
print column 132, "!"
if bctc = 1 then begin
  if class = "C" then print "!", hipri_pri using "##### ", "!";
  else print "!" ";
end
else print "!", prior1_score using "##### ", "!";
if ssnd_def1>0 then print ssnd_def1 using " #####", "-", ssnd_con1;
if ssnd_def2>0 then print ssnd_def2 using " #####", "-", ssnd_con2;
if ssnd_def3>0 then print ssnd_def3 using " #####", "-", ssnd_con3;
if ssnd_def4>0 then print ssnd_def4 using " #####", "-", ssnd_con4;
if ssnd_def5>0 then print ssnd_def5 using " #####", "-", ssnd_con5;
if ssnd_def6>0 then print ssnd_def6 using " #####", "-", ssnd_con6;
if ssnd_def7>0 then print ssnd_def7 using " #####", "-", ssnd_con7;
print column 71, "!", column 79, "!", column 132, "!"
let lcnt = lcnt + 3

before group of wkpkg_no
if wkpkg_pe <> " " then begin
  if lcnt > 45 then begin
    let cont = 1
    print "!", column 9, "!", column 71, "!", column 79, "!", column 132, "!"
    print "-----"
    "-----";
    skip to top of page
  end
  if head = 1 then begin
    print "-----"
    "-----"
    print "!" SSN "!", 6 spaces, "TITLE", column 58, "DA/TDC MA",
      column 71, "!" PDIP "!",
      " 87 88 89 90 91 92 93 !"
  end
end

```

```

print "|-----|-----";
"-----";
let lcnt = lcnt + 3
let head = 0
end
let cnt = cnt + 1
let rf1 = rf1 + wkpfr_f1
let rf2 = rf2 + wkpfr_f2
let rf3 = rf3 + wkpfr_f3
let rf4 = rf4 + wkpfr_f4
let rf5 = rf5 + wkpfr_f5
let rf6 = rf6 + wkpfr_f6
let rf7 = rf7 + wkpfr_f7
print "|", column 9, "!", 1 spaces, cnt, ". ", 2 spaces, wkpkg_pe,
      2 spaces, wkpkg_proj, 2 spaces, wkpkg_task clipped, 2 spaces,
      wkpkg_no clipped, 2 spaces, wkpkg_cmd clipped, " Crit:",
      uniwkp_srf using " ##", column 71, "!", column 79, "!" ";
if wkpfr_f1>0 then print wkpfr_f1 using "##### "; else print " ";
if wkpfr_f2>0 then print wkpfr_f2 using "##### "; else print " ";
if wkpfr_f3>0 then print wkpfr_f3 using "##### "; else print " ";
if wkpfr_f4>0 then print wkpfr_f4 using "##### "; else print " ";
if wkpfr_f5>0 then print wkpfr_f5 using "##### "; else print " ";
if wkpfr_f6>0 then print wkpfr_f6 using "##### "; else print " ";
if wkpfr_f7>0 then print wkpfr_f7 using "##### "; else print " ";
print column 132, "!"
print "|", column 9, "!", 10 spaces, wkpkg_title[1, 50], column 71, "!",
      wkpkg_pdip[1, 4], "-", wkpkg_pdip[5, 6], "!" ";
if wkpfr_u1>0 then print -wkpfr_u1 using "((((#)"; else print " ";
if wkpfr_u2>0 then print -wkpfr_u2 using "((((#)"; else print " ";
if wkpfr_u3>0 then print -wkpfr_u3 using "((((#)"; else print " ";
if wkpfr_u4>0 then print -wkpfr_u4 using "((((#)"; else print " ";
if wkpfr_u5>0 then print -wkpfr_u5 using "((((#)"; else print " ";
if wkpfr_u6>0 then print -wkpfr_u6 using "((((#)"; else print " ";
if wkpfr_u7>0 then print -wkpfr_u7 using "((((#)"; else print " ";
print column 132, "!"
let lcnt = lcnt + 2
end

after group of scl_ssn
if lcnt > 44 then begin
let cnt = 1
print "|", column 9, "!", column 71, "!", column 79, "!", column 132, "!"
print "-----";
"-----";
skip to top of page
end
if head = 1 then begin
print "-----";
"-----";
print "|: SSN |", 6 spaces, "TITLE", column 58, "DA/TDC MA",
      column 71, ":", PDIP |",
      " 87 88 89 90 91 92 93 |"
print "|-----|-----";
"-----";
let lcnt = lcnt + 3
let head = 0

```

```
end
print ":",column 9,"!",column 71,"!",column 79,"!",column 132,"!"
print ":",column 9,"!",21 spaces,"Cumulative Totals: ",
column 50,"Procurement Funded: ",column 71,"!",column 79,"! ",
pf1 using "##### ",pf2 using "##### ",pf3 using "##### ",
pf4 using "##### ",pf5 using "##### ",pf6 using "##### ",
pf7 using "##### ",column 132,"!"
print ":",column 9,"!",column 57,"RDTE Funded: ",column 71,"!",
column 79,"! ",
rf1 using "##### ",rf2 using "##### ",rf3 using "##### ",
rf4 using "##### ",rf5 using "##### ",rf6 using "##### ",
rf7 using "##### ",column 132,"!"
let lcnt = lcnt + 3
end
```

< Pdip to system cross reference report. Moved to the mamp database on 5/2/86.
6/11/86 changed header info somewhat.
6/13/86 revised pdip format, added titles and priorities. >

database mamp end

define

```
param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12
variable pno type integer
variable st type integer
variable ll type integer
```

end

input

```
prompt for pno using "Please enter the starting page number > "
```

end

output

```
left margin 0
right margin 132
report to "pdipsys.out"
```

end

read into a

```
scl_ssn
where scl_code = code and scl_name = name
```

end

read into b

```
a
ssn_anc_mgr ssn_da_ma ssn_tradoc_ma ssn_tradoc_pro
lrrdp_title lrrdp_pdip
joining a.scl_ssn = optional ssn_ssn
and a.scl_ssn = optional lrrdp_ssn
```

end

read into c

```
b
lrrdppri_pri
lrrdppri_title lrrdppri_tradoc_pro lrrdppri_tradoc_ma lrrdppri_da_ma
joining b.lrrdp_pdip = optional lrrdppri_pdip
```

end

sort by lrrdp_pdip[1,4] lrrdp_pdip[5,6] scl_ssn end

format

before group of lrrdp_pdip[1,4]

```
if ll<5 then skip to top of page
skip 2 lines
if lrrdp_pdip[1,4]=" " then print "____";
else print lrrdp_pdip[1,4];
if lrrdppri_da_ma=" " then print 32 spaces, "____";
else print 32 spaces, lrrdppri_da_ma;
```

Jun 13 09:05 1986 pdipsys Page 2

```

if lrrdppri_tradoc_ma=" " then print 5 spaces, "___";
else print 5 spaces, lrrdppri_tradoc_ma;
if lrrdppri_tradoc_pro=" " then print 3 spaces, "___";
else print 3 spaces, lrrdppri_tradoc_pro;
if lrrdppri_title=" " then print 9 spaces, "_____";
else print 9 spaces, lrrdppri_title
let ll = ll-3

before group of lrrdp_pdip[5,6]
if ll<2 then skip to top of page
if lrrdp_pdip[5,6]=" " then print 8 spaces, "___";
else print 8 spaces, lrrdp_pdip[5,6];
if lrrdppri_pri = 0.0 then print 4 spaces, "Not Available"
else print 4 spaces, lrrdppri_pri using "####.##"
let ll = ll-1

before group of scl_ssn
print 26 spaces, scl_ssn, 4 spaces, ssn_da_ma, 5 spaces, ssn_tradoc_ma, " ",
      ssn_tradoc_pro, 4 spaces, lrrdp_title[1,50], 4 spaces, ssn_amc_mgr
let ll = ll-1

page header
print column 47, "***** C O N F I D E N T I A L *****"
skip 2 lines
print column 49, "PDIP TO ", name clipped, " SYSTEM CROSS REFERENCE"
skip 2 lines
print "PDIP", 3 spaces, "INCR", 3 spaces, "PRIORITY", 5 spaces,
      "SSN", 5 spaces, "DA MA", 3 spaces, "TDC MA/PROP", 16 spaces,
      "PDIP/SYSTEM TITLE", 24 spaces, "COMMAND", 11 spaces
skip 1 line
let ll = 47
let st = 1

page trailer
skip 2 lines
print column 47, "***** C O N F I D E N T I A L *****"
print column 100, date, 2 spaces, time
if pletter="Z" then print 60 spaces, pno
else print 60 spaces, pletter, "-", pno using "###"
let pno = pno+1

end

```

(POM 89 REPORT OF PDIPS WITH CUMULATIVE RDTE FUNDING SHOWN.

6/23/86. taken from laura's pdipwkp report. This differs in that it works with the projected pom89 pdip alignments and priorities.
6/24/86. changed pom 89 pdip structure. added pdip parameter.
7/10/86. changed the pom89ssn file structure.
7/15/86. changed uniwkp_wc to uniwkp_srf.)

database mamp end

define

variable lcnt type integer
variable cnt type integer
variable head type integer
variable cont type integer
variable break type integer
variable obreak type integer
variable pno type integer
variable tot type float

variable pf1 type long
variable pf2 type long
variable pf3 type long
variable pf4 type long
variable pf5 type long
variable pf6 type long
variable pf7 type long

variable rf1 type long
variable rf2 type long
variable rf3 type long
variable rf4 type long
variable rf5 type long
variable rf6 type long
variable rf7 type long

param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12
param[4] pdip type character length 6

end

input

prompt for pno using "Please enter the starting page number > "
end

output

left margin 0
right margin 132
report to "pom89cum.out"

end

read into b

scl_ssn
pom89ssn_sys_idx pom89ssn_seq pom89ssn_pdip
where scl_name = name and scl_code = code and pom89ssn_pdip matches pdip

```
      joining scl_ssn = pom89ssn_ssn
    end

    read into a
      b
      ssn_anc_mgr ssn_acq_code ssn_tradoc_ma ssn_da_ma
      ssndroll
      joining b. scl_ssn = ssn_ssn
        and b. scl_ssn = ssnd_ssn
    end

    read into b
      a
      pom89
      lrrdp_title lrrdp_procf2 lrrdp_procf3 lrrdp_procf4 lrrdp_procf5
      lrrdp_procf6 lrrdp_procf7 lrrdp_procf8
      joining a.pom89ssn_pdip = optional pom89_pdip
        and a.pom89ssn_sys_idx = optional lrrdp_sys_idx
    end

    read into d
      pcl_idx
      where pcl_code = code and pcl_name = name and pcl_cat = "6.3"
        and pcl_subcat <> "6.5"
    end

    read into e
      d
      wkpkg_wkpsys_idx wkpkg_title
      wkpfr_f0 wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
      wkpfr_u0 wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
      joining d.pcl_idx = wkpkg_proj_idx
        and wkpkg_no_idx = optional wkpfr_wkpkg_idx
    end

    read into c
      b
      e
      uniwp_srf uniwp_seq uniwp_pom89
      joining b.pom89ssn_sys_idx = optional uniwp_sys_idx
        and uniwp_wkpkg_idx = optional e.wkpkg_wkpsys_idx
    end

    sort by pom89ssn_pdip
      pom89ssn_seq
      scl_ssn
      uniwp_seq
      wkpkg_no
    end

    format

    page header
      print column 47, "***** C O N F I D E N T I A L *****"
      skip 2 lines
      print column 49, pom89_title;
```

Jul 15 09:18 1986 pom89cum Page 3

```
print 4 spaces, "(" , pom89ssn_pdip[1,4], "-", pom89ssn_pdip[5,6], ")"
print column 47, "POM 89 PDIP-INCREMENT BREAKOUT FOR ", name clipped
print column 49, "WITH CUMULATIVE TOTALS (RTDE/PROC)";
if cont = 1 then print " (continued)" else print ""
let cont = 0
skip 2 lines
let head = 1
let lcnt = 0
```

```
page trailer
print column 47, "***** C O N F I D E N T I A L *****"
print column 100, date, 2 spaces, time
if pletter="Z" then print column 60, pno
else print column 60, pletter, "-", pno
using "###" let pno = pno+1
```

```
before group of pom89ssn_pdip
let pf1 = 0
let pf2 = 0
let pf3 = 0
let pf4 = 0
let pf5 = 0
let pf6 = 0
let pf7 = 0
let rf1 = 0
let rf2 = 0
let rf3 = 0
let rf4 = 0
let rf5 = 0
let rf6 = 0
let rf7 = 0
skip to top of page
print "Acquisition Strategy: ";
let break = 110
while pom89_strat[break, break]<>" " do let break = break-1
print pom89_strat[1, break]
let obreak = break+1
let break = break+125
while pom89_strat[break, break]<>" " do let break = break-1
print 7 spaces, pom89_strat[obreak, break]
let obreak = break+1
let break = break+125
while pom89_strat[break, break]<>" " do let break = break-1
print 7 spaces, pom89_strat[obreak, break]
let obreak = break+1
let break = break+125
while pom89_strat[break, break]<>" " do let break = break-1
print 7 spaces, pom89_strat[obreak, break]
let obreak = break+1
let break = break+125
if break>600 then let break = 600
while pom89_strat[break, break]<>" " do let break = break-1
print 7 spaces, pom89_strat[obreak, break]
skip 1 lines
let lcnt = lcnt + 7
```



```
after group of pom89ssn_p dip
print "!", column 9, "!", column 71, "!", column 79, "!", column 132, "!"
print "-----",
      "-----"

before group of scl_ssn
let cnt = 0
if lcnt > 42 then begin
  let cont = 1
  print "!", column 9, "!", column 71, "!", column 79, "!", column 132, "!"
  print "-----",
        "-----"

  skip to top of page
end
if head = 1 then begin
  print "-----",
        "-----"

  print "!", SSN "!", 6 spaces, "TITLE", column 58, "DA/TDC MA",
        column 71, "!", PDIP "!",
        " 87 88 89 90 91 92 93 !"
  print "!"-----"!-----",
        "-----;"

  let lcnt = lcnt + 3
  let head = 0
end
print "!", column 9, "!", column 71, "!", column 79, "!", column 132, "!"
print "!", scl_ssn, column 9, "!", 1 space, lrrdp_title[1,40], 6 space,
      ssn_da_ma, " / ", ssn_tradoc_ma, column 71, "!", pom89ssn_p dip[1,4], "-",
      pom89ssn_p dip[5,6], "!", column 81;
let pf1 = pf1 + lrrdp_proc f2
let pf2 = pf2 + lrrdp_proc f3
let pf3 = pf3 + lrrdp_proc f4
let pf4 = pf4 + lrrdp_proc f5
let pf5 = pf5 + lrrdp_proc f6
let pf6 = pf6 + lrrdp_proc f7
let pf7 = pf7 + lrrdp_proc f8
let tot = lrrdp_proc f2 + lrrdp_proc f3 + lrrdp_proc f4 + lrrdp_proc f5 +
          lrrdp_proc f6 + lrrdp_proc f7 + lrrdp_proc f8
if tot = 0 then print column 90, "(Procurement Funds Not Scheduled)";
else begin
  if lrrdp_proc f2 > 0 then print lrrdp_proc f2 using "##### ";
  else print " ";
  if lrrdp_proc f3 > 0 then print lrrdp_proc f3 using "##### ";
  else print " ";
  if lrrdp_proc f4 > 0 then print lrrdp_proc f4 using "##### ";
  else print " ";
  if lrrdp_proc f5 > 0 then print lrrdp_proc f5 using "##### ";
  else print " ";
  if lrrdp_proc f6 > 0 then print lrrdp_proc f6 using "##### ";
  else print " ";
  if lrrdp_proc f7 > 0 then print lrrdp_proc f7 using "##### ";
  else print " ";
  if lrrdp_proc f8 > 0 then print lrrdp_proc f8 using "##### ";
  else print " ";
end
print column 132, "!"
```

```

print "!", pom89ssn_seq using " ##### ", "!";
if ssnd_def1>0 then print ssnd_def1 using " #####", "-", ssnd_con1;
if ssnd_def2>0 then print ssnd_def2 using " #####", "-", ssnd_con2;
if ssnd_def3>0 then print ssnd_def3 using " #####", "-", ssnd_con3;
if ssnd_def4>0 then print ssnd_def4 using " #####", "-", ssnd_con4;
if ssnd_def5>0 then print ssnd_def5 using " #####", "-", ssnd_con5;
if ssnd_def6>0 then print ssnd_def6 using " #####", "-", ssnd_con6;
if ssnd_def7>0 then print ssnd_def7 using " #####", "-", ssnd_con7;
print column 71, "!", column 79, "!", column 132, "!"
let lcnt = lcnt + 3

```

```

before group of wkpkg_no
if wkpkg_pe <> " " then begin
  if lcnt > 45 then begin
    let cnt = 1
    print "!", column 9, "!", column 71, "!", column 79, "!", column 132, "!"
    print "-----",
    "-----"
    skip to top of page
  end
  if head = 1 then begin
    print "-----",
    "-----"
    print ":", SSN ":", 6 spaces, "TITLE", column 58, "DA/TDC MA",
    column 71, ":", PDIP ":",
    " 87 88 89 90 91 92 93 :",
    print "!"-----"!-----",
    "-----!"
    let lcnt = lcnt + 3
    let head = 0
  end
  let cnt = cnt + 1
  let rf1 = rf1 + wkpfr_f1
  let rf2 = rf2 + wkpfr_f2
  let rf3 = rf3 + wkpfr_f3
  let rf4 = rf4 + wkpfr_f4
  let rf5 = rf5 + wkpfr_f5
  let rf6 = rf6 + wkpfr_f6
  let rf7 = rf7 + wkpfr_f7
  print "!", column 9, "!", 1 spaces, cnt, " ", 2 spaces, wkpkg_pe,
  2 spaces, wkpkg_proj, 2 spaces, wkpkg_task clipped, 2 spaces,
  wkpkg_no clipped, 2 spaces, wkpkg_cmd clipped, " Crit:",
  uniwp_srf using " ##", column 71, "!", column 79, "!" ;
  if wkpfr_f1>0 then print wkpfr_f1 using "##### "; else print " ";
  if wkpfr_f2>0 then print wkpfr_f2 using "##### "; else print " ";
  if wkpfr_f3>0 then print wkpfr_f3 using "##### "; else print " ";
  if wkpfr_f4>0 then print wkpfr_f4 using "##### "; else print " ";
  if wkpfr_f5>0 then print wkpfr_f5 using "##### "; else print " ";
  if wkpfr_f6>0 then print wkpfr_f6 using "##### "; else print " ";
  if wkpfr_f7>0 then print wkpfr_f7 using "##### "; else print " ";
  print column 132, "!"
  print "!", column 9, "!", 10 spaces, wkpkg_title[1, 50], column 71, "!",
  uniwp_pom89[1, 4], "-", uniwp_pom89[5, 6], "!" ;
  if wkpfr_u1>0 then print -wkpfr_u1 using "((( (#)"; else print " ";
  if wkpfr_u2>0 then print -wkpfr_u2 using "((( (#)"; else print " ";
  if wkpfr_u3>0 then print -wkpfr_u3 using "((( (#)"; else print " ";

```

```

if wkpfr_u4>0 then print -wkpfr_u4 using "(((("#); else print " ";
if wkpfr_u5>0 then print -wkpfr_u5 using "(((("#); else print " ";
if wkpfr_u6>0 then print -wkpfr_u6 using "(((("#); else print " ";
if wkpfr_u7>0 then print -wkpfr_u7 using "(((("#); else print " ";
print column 132,"!"
let lcnt = lcnt + 2
end

after group of scl_ssn
if lcnt > 44 then begin
let cont = 1
print "!",column 9,"!",column 71,"!",column 79,"!",column 132,"!"
print "-----",
"-----"
skip to top of page
end
if head = 1 then begin
print "-----",
"-----"
print "!" SSN "!",6 spaces,"TITLE",column 58,"DA/TDC MA",
column 71,"! PDIP "!,
" 87 88 89 90 91 92 93 !"
print "!"-----"!
"-----!"
let lcnt = lcnt + 3
let head = 0
end
print "!",column 9,"!",column 71,"!",column 79,"!",column 132,"!"
print "!",column 9,"!",21 spaces,"Cumulative Totals: ",
column 50,"Procurement Funded: ",column 71,"!",column 79,"! ",
pf1 using "***** ",pf2 using "***** ",pf3 using "***** ",
pf4 using "***** ",pf5 using "***** ",pf6 using "***** ",
pf7 using "***** ",column 132,"!"
print "!",column 9,"!",column 57,"RDTE Funded: ",column 71,"!",
column 79,"! ",
rf1 using "***** ",rf2 using "***** ",rf3 using "***** ",
rf4 using "***** ",rf5 using "***** ",rf6 using "***** ",
rf7 using "***** ",column 132,"!"
let lcnt = lcnt + 3

end

```

Jun 3 14 31 1986 priori Page 1

(System Priority Ratings Developer)
(4/23/86 changed to mamp database
5/2/86 added the blank count field. added the unique ssn check.)

database mamp end

define

variable tot type integer
variable ndef type integer
variable na type integer
variable nb type integer
variable nc type integer
variable nd type integer
variable ne type integer
variable nx type integer
variable bl type integer
variable score type float
variable conval type float
variable defval type float

end

output

page length 32000
left margin 0
right margin 80
top margin 0
report to "prior1.out"

end

read

ssndef_ssn ssndef_def ssndef_cont_value
joining unissn_ssn = ssndef_ssn

end

sort by ssndef_ssn ssndef_def ssndef_cont_value end

format

before group of ssndef_ssn

let score = 0.0
let ndef = 0
let na = 0
let nb = 0
let nc = 0
let nd = 0
let ne = 0
let nx = 0
let bl = 0

before group of ssndef_def

if ssndef_def > 0 then begin
if ssndef_cont_value = "A" then begin
let na = na+1
let conval = 16.0 end
else if ssndef_cont_value = "B" then begin
let nb = nb+1

Jun 3 14:31 1986 prior1 Page 2

```
    let conval = 8.0 end
  else if ssndef_cont_value = "C" then begin
    let nc = nc+1
    let conval = 4.0 end
  else if ssndef_cont_value = "D" then begin
    let nd = nd+1
    let conval = 2.0 end
  else if ssndef_cont_value = "E" then begin
    let ne = ne+1
    let conval = 1.0 end
  else if ssndef_cont_value = "F" then begin
    let nx = nx+1
    let conval = 0.0 end
  else begin
    let bl = bl+1
    let conval = 0.0
  end
  if ssndef_def = 1001 then let defval = 30.0
  else let defval = (500.0-ssndef_def)/10.0
  let score = score+defval*conval
  let ndef = ndef+1
end
after group of ssndef_ssn
if ndef>0 then begin
  print ssndef_ssn, ";", ndef, ";", na, ";", nb, ";", nc, ";", nd,
    ";", ne, ";", nx, ";", bl, ";", score, ";"
end
end
```

Jul 10 13:14 1986 prior2 Page 1

```
< Workpackage Priority Ratings Generator >
< 4/23/86 changed to the mamp data base
  5/2/86 changed the blank workpackage criticality value to zero. added the
    unique ssn check.
  7/10/86 changed the workpackage criticality factor from wkpsys_srf to
    wkpsys_srf in keeping with iitri's usage.>
```

database mamp end

```
define
  variable nsys type integer
  variable hisys type float
  variable losys type float
  variable score type float
  variable conval type float
end
```

```
output
  page length 32000
  top margin 0
  left margin 0
  right margin 132
  report to "prior2.out"
end
```

```
read into a
  wkpsys_cmd wkpsys_cat wkpsys_wkpkg wkpsys_ssn wkpsys_srf
  priori_score
  joining unissn_ssn = wkpsys_ssn
    and unissn_ssn = priori_ssn_no
end
```

sort by wkpsys_cmd wkpsys_wkpkg wkpsys_ssn end

format

```
before group of wkpsys_wkpkg
  let score = 0.0
  let nsys = 0
  let hisys = 0.0
  let losys = 100000.0
```

```
before group of wkpsys_ssn
  if wkpsys_ssn <> " " then begin
    if wkpsys_srf = 1 then let conval = 10.0
    else if wkpsys_srf = 2 then let conval = 5.0
    else if wkpsys_srf = 3 then let conval = 2.5
    else if wkpsys_srf = 4 then let conval = 1.25
    else let conval = 0.0
    let score = score+priori_score*conval
    let nsys = nsys+1
    if priori_score>hisys then let hisys = priori_score
    if priori_score<losys then let losys = priori_score
  end
```

after group of wkpsys_wkpkg

Jul 10 13:14 1986 prior2 Page 2

```
if nsys=0 then let losys = 0.0
print wkpsys_cmd, ":", wkpsys_cat, ":", wkpsys_wkpkg, ":", nsys, ":",
      hisys, ":", losys, ":", score, ":",
```

end

```
( System Priority Ratings Report )  
( Previously computed Ratings stored in Prior1 )  
( This version prints out the unclassified version with just the ratings and  
  a generic tabulation of the number of ratings of each category.  
  4/23/86 switch to the mamp data base.  
  5/5/86 added #blanks to the output.  
  5/12/86 added base case systems at the top of the list. )
```

```
database mamp end
```

```
define  
  variable cnt      type integer  
  variable pno      type integer  
  param[1] pletter  type character length 1  
  param[2] code     type integer  
  param[3] name     type character length 12  
end
```

```
input  
  prompt for pno using "Please enter the starting page number > "  
end
```

```
output  
  left margin 0  
  right margin 132  
  report to "priorsys.out"  
end
```

```
read into b  
  scl_ssn  
  where scl_code = code and scl_name = name  
end
```

```
read into c  
  b  
  ssn_amc_mgr ssn_da_ma  
  bc = 0  
  where ssn_acq_code > 2 or ssn_acq_code = 0  
  joining b.scl_ssn = optional ssn_ssn  
end
```

```
read into d  
  b  
  ssn_amc_mgr ssn_da_ma  
  bc = 1  
  where ssn_acq_code = 1  
  joining b.scl_ssn = ssn_ssn  
end
```

```
assign e = c Union d end
```

```
read into a  
  e  
  lrrdp_title  
  priori  
  joining e.scl_ssn = optional priori_ssn_no
```



```
        and e.scl_ssn = optional lrrdp_ssn
end

sort by bc descending priori_score descending priori_ndef descending
       scl_ssn end

format

page header
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 46, name clipped, " 1-N DEVELOPMENT SYSTEM BDP RANKING"
  skip 2 lines
  print 7 spaces, "SSN", 5 spaces, "DA MA", 4 spaces, "AMC MGR", 15 spaces,
    "TITLE", column 88, " #DEF #A #B #C #D #E #X BL RATING"
  skip 1 line

page trailer
  skip 2 lines
  print column 47, "***** U N C L A S S I F I E D *****"
  print column 100, date, 2 spaces, time
  if pletter="Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1

before group of scl_ssn
  let cnt = cnt+1
  print cnt using "###. ", scl_ssn, 4 spaces, ssn_da_ma,
    5 spaces, ssn_amc_mgr, column 30, lrrdp_title[1, 50], 2 spaces,
    priori_ndef using "###", priori_na using "###", priori_nb using "###",
    priori_nc using "###", priori_nd using "###",
    priori_ne using "###", priori_nx using "###", priori_bl using "###";
  if bc = 1 then print " base case"
  else print priori_score using "#####.##"

end
```

```
( Tech Base Workpackage Priority Ratings Report )
( Previously Computed Priorities from Prior1 and Prior2.
  3/21/86 taken from priorwp
  this report allows only 6.1, 6.2, and 6.3A workpackages.
  workpackages are selected which are linked in any way to systems
  in the ssncontrol file. )
```

```
database mamp end
```

```
define
  variable cnt      type integer
  variable pno      type integer
  param[1] pletter  type character length 1
  param[2] code     type integer
  param[3] name     type character length 12
end
```

```
input
  prompt for pno using "Please enter the starting page number > "
end
```

```
output
  left margin 0
  right margin 132
  report to "priortb.out"
end
```

```
read into b
  scl_ssn
  where scl_code = code and scl_name = name
end
```

```
read into a
  wkpkg_cmd wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no wkpkg_title
  prior2_score prior2_nsys
  where (wkpkg_subcat="6.1" or wkpkg_subcat="6.2" or wkpkg_subcat="6.3A")
  joining b.scl_ssn = wkpsys_ssn
         and wkpsys_wkpkg_idx = wkpkg_wkpsys_idx
         and wkpkg_no_idx = optional prior2_wkpkg_idx
end
```

```
sort by prior2_score descending prior2_nsys descending
       wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no end
```

```
format
```

```
page header
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 49, "BDP TECH BASE WORKPACKAGE RANKINGS"
  print column 49, "          FOR ", name
  skip 2 lines
  print 6 spaces, "PE    PROJ TASK    NO", 18 spaces, "TITLE", 45 spaces,
  "COMMAND", 10 spaces, "RATING # SYS  "
  skip 1 line
```

Jun 3 14:31 1986 priortb Page 2

```
page trailer
  skip 2 lines
  print column 47, "***** U N C L A S S I F I E D *****"
  print column 100, date, 2 spaces, time
  if pletter="Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1

after group of wkpkg_no
  let cnt = cnt+1
  print cnt using "###. ", wkpkg_pe, 1 space, wkpkg_proj, 1 space, wkpkg_task,
    1 space, wkpkg_no, 3 spaces, wkpkg_title, wkpkg_cmd, prior2_score,
    1 space, prior2_nsys

end
```

Jun 3 14.31 1986 priorwp Page 1

```
( Workpackage Priority Ratings Report )
( Previously Computed Priorities from Prior1 and Prior2.
  4/23/86 changed to mamp data base.
  5/12/86 limited to 6.3, 6.4, and 6.7 workpackages. )
```

database mamp end

define

```
variable cnt      type integer
variable pno      type integer
param[1] pletter  type character length 1
param[2] code     type integer
param[3] name     type character length 12
```

end

input

```
prompt for pno using "Please enter the starting page number > "
end
```

output

```
left margin 0
right margin 132
report to "priorwp.out"
```

end

read into b

```
pcl_idx
where pcl_code = code and pcl_name = name
and (pcl_subcat = "6.3A" or pcl_subcat = "6.3B" or pcl_subcat = "6.4"
or pcl_subcat = "6.7")
```

end

read into a

```
wkpkg_cmd wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no wkpkg_title
prior2_score prior2_nsys
joining b.pcl_idx = wkpkg_proj_idx
and wkpkg_no_idx = optional prior2_wkpkg_idx
```

end

```
sort by prior2_score descending prior2_nsys descending
wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no end
```

format

page header

```
print column 47, "***** U N C L A S S I F I E D *****"
skip 2 lines
print column 49, name clipped, " 1-N WORKPACKAGE BDP RANKINGS"
skip 2 lines
print 6 spaces, "PE      PROJ TASK      NO", 18 spaces, "TITLE", 45 spaces,
"CGMMAND", 10 spaces, "RATING # SYS      "
skip 1 line
```

page trailer

```
skip 2 lines
print column 47, "***** U N C L A S S I F I E D *****"
```

Jun 3 14:31 1986 priorwp Page 2

```
print column 100, date, 2 spaces, time
if pletter="Z" then print column 60, pno
else print column 60, pletter, "-", pno using "###"
let pno = pno+1

after group of wkpkg_no
let cnt = cnt+1
print cnt using "####. ", wkpkg_pe, 1 space, wkpkg_proj, 1 space, wkpkg_task,
1 space, wkpkg_no, 3 spaces, wkpkg_title, wkpkg_cmd, prior2_score,
1 space, prior2_nsys

end
```

{ BDP priority system decision aid. Shows systems in BDP priority, base case first, and shows uniquely linked workpackages, with cumulative rdte funding after each system with rdte linked. }

database mamp end

define

variable lcnt type integer
variable cnt type integer
variable head type integer
variable break type integer
variable obreak type integer
variable pno type integer
variable tot type float

variable f1 type long
variable f2 type long
variable f3 type long
variable f4 type long
variable f5 type long
variable f6 type long
variable f7 type long

variable u1 type long
variable u2 type long
variable u3 type long
variable u4 type long
variable u5 type long
variable u6 type long
variable u7 type long

param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12

end

input

prompt for pno using "Please enter the starting page number > "
end

output

left margin 0
right margin 132
report to "prisysda2.out"
end

read into b

scl_ssn
where scl_name = name and scl_code = code
end

read into a

b
ssn_anc_mgr ssn_tradoc_ma ssn_da_ma
bc = 1
where ssn_acq_code = 1

```
    joining b.scl_ssn = ssn_ssn
end

read into d
  b
  ssn_anc_mgr ssn_tradoc_ma ssn_da_ma
  bc = 0
  where ssn_acq_code > 2
  joining b.scl_ssn = ssn_ssn
end

assign c = a union d end

read into a
  c
  priori_score
  hipri_title
  joining c.scl_ssn = optional priori_ssn_no
    and c.scl_ssn = optional hipri_ssn
end

read into d
  pcl_idx
  where pcl_code = code and pcl_name = name
end

read into e
  d
  wkpkg_wkpsys_idx wkpkg_title
  wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
  wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
  joining d.pcl_idx = wkpkg_proj_idx
    and wkpkg_no_idx = optional wkpfr_wkpkg_idx
end

read into c
  a
  e
  uniwp_srf
  joining a.scl_ssn = optional uniwp_ssn
    and uniwp_wkpkg_idx = optional e.wkpkg_wkpsys_idx
end

sort by bc descending
  priori_score descending
  scl_ssn
  wkpkg_proj
  wkpkg_no
end

format

page header
print column 47, "***** U N C L A S S I F I E D *****"
skip 2 lines
print column 46, "BDP SYSTEM RANKING DECISION AID FOR ", name clipped
```

```
print column 50, "WITH CUMULATIVE TOTALS (RTDE)"
skip 2 lines
let head = 1
let lcnt = 0

page trailer
print column 47, "***** U N C L A S S I F I E D *****"
print column 100, date, 2 spaces, time
if pletter="Z" then print column 60, pno
else print column 60, pletter, "-", pno using "###"
let pno = pno+1

on last record
print ":", column 9, ":", column 71, ":", column 79, ":", column 132, "!"
print "-----";

before group of scl_ssn
let cnt = 0
if lcnt > 42 then begin
print ":", column 9, ":", column 71, ":", column 79, ":", column 132, "!"
print "-----";

skip to top of page
end
if head = 1 then begin
print "-----";
print ":", SSN ":", 6 spaces, "TITLE", column 58, "DA/TDC MA",
column 71, ":", BDP ":",
" 87 88 89 90 91 92 93 !"
print "-----";
let lcnt = lcnt + 3
let head = 0
end
print ":", column 9, ":", column 71, ":", column 79, ":", column 132, "!"
print ":", scl_ssn, ":", hipri_title[1, 45], 2 spaces, ssn_da_ma, " / ",
ssn_tradoc_ma, column 71, ":", priori_score using "#### ", "!",
column 132, "!"
let lcnt = lcnt + 2

before group of wkpkg_no
if wkpkg_pe <> " " then begin
if lcnt > 45 then begin
print ":", column 9, ":", column 71, ":", column 79, ":", column 132, "!"
print "-----";

skip to top of page
end
if head = 1 then begin
print "-----";
print ":", SSN ":", 6 spaces, "TITLE", column 58, "DA/TDC MA",
column 71, ":", BDP ":",
" 87 88 89 90 91 92 93 !"
print "-----";

```



```

print " |-----|-----";
"-----";
let lcnt = lcnt + 3
let head = 0
end
let cnt = cnt + 1
let f1 = f1 + wkpfr_f1
let f2 = f2 + wkpfr_f2
let f3 = f3 + wkpfr_f3
let f4 = f4 + wkpfr_f4
let f5 = f5 + wkpfr_f5
let f6 = f6 + wkpfr_f6
let f7 = f7 + wkpfr_f7
let u1 = u1 + wkpfr_u1
let u2 = u2 + wkpfr_u2
let u3 = u3 + wkpfr_u3
let u4 = u4 + wkpfr_u4
let u5 = u5 + wkpfr_u5
let u6 = u6 + wkpfr_u6
let u7 = u7 + wkpfr_u7
print "!", column 9, "!", 1 spaces, cnt, ".", 2 spaces, wkpkg_pe,
2 spaces, wkpkg_proj, 2 spaces, wkpkg_task clipped, 2 spaces,
wkpkg_no clipped, 2 spaces, wkpkg_cmd clipped, column 58, "Crit:",
uniwkp_srf using "#", column 71, "!", column 79, "!" ";
if wkpfr_f1>0 then print wkpfr_f1 using "##### "; else print " ";
if wkpfr_f2>0 then print wkpfr_f2 using "##### "; else print " ";
if wkpfr_f3>0 then print wkpfr_f3 using "##### "; else print " ";
if wkpfr_f4>0 then print wkpfr_f4 using "##### "; else print " ";
if wkpfr_f5>0 then print wkpfr_f5 using "##### "; else print " ";
if wkpfr_f6>0 then print wkpfr_f6 using "##### "; else print " ";
if wkpfr_f7>0 then print wkpfr_f7 using "##### "; else print " ";
print column 132, "!"
print "!", column 9, "!", 10 spaces, wkpkg_title[1, 50], column 71, "!",
column 79, "!" ";
if wkpfr_u1>0 then print -wkpfr_u1 using "(((((#)"; else print " ";
if wkpfr_u2>0 then print -wkpfr_u2 using "(((((#)"; else print " ";
if wkpfr_u3>0 then print -wkpfr_u3 using "(((((#)"; else print " ";
if wkpfr_u4>0 then print -wkpfr_u4 using "(((((#)"; else print " ";
if wkpfr_u5>0 then print -wkpfr_u5 using "(((((#)"; else print " ";
if wkpfr_u6>0 then print -wkpfr_u6 using "(((((#)"; else print " ";
if wkpfr_u7>0 then print -wkpfr_u7 using "(((((#)"; else print " ";
print column 132, "!"
let lcnt = lcnt + 2
end
after group of scl_ssn
if cnt > 0 then begin
if lcnt > 44 then begin
print "!", column 9, "!", column 71, "!", column 79, "!", column 132, "!"
print " |-----|-----";
"-----";
skip to top of page
end
if head = 1 then begin
print " |-----|-----";
"-----";

```

```
print "! SSN !", 6 spaces, "TITLE", column 58, "DA/TDC MA",
      column 71, "! BDP !",
      "      87      88      89      90      91      92      93      !"
print "!-----|-----|-----|-----|-----|-----|-----|",
      "-----|-----|-----|-----|-----|-----|-----|"
let lcnt = lcnt + 3
let head = 0
end
print "!", column 9, "!", column 71, "!", column 79, "!", column 132, "!"
print "!", column 9, "!", 21 spaces, "Cumulative Totals: ",
      column 50, "RDTE Funded: ", column 71, "!", column 79, "!",
      f1 using "***** ", f2 using "***** ", f3 using "***** ",
      f4 using "***** ", f5 using "***** ", f6 using "***** ",
      f7 using "***** ", column 132, "!"
print "!", column 9, "!",
      column 50, "RDTE Unfunded: ", column 71, "!", column 79, "!",
      u1 using "***** ", u2 using "***** ", u3 using "***** ",
      u4 using "***** ", u5 using "***** ", u6 using "***** ",
      u7 using "***** ", column 132, "!"
let lcnt = lcnt + 3
end
end
```

AD-A172 652

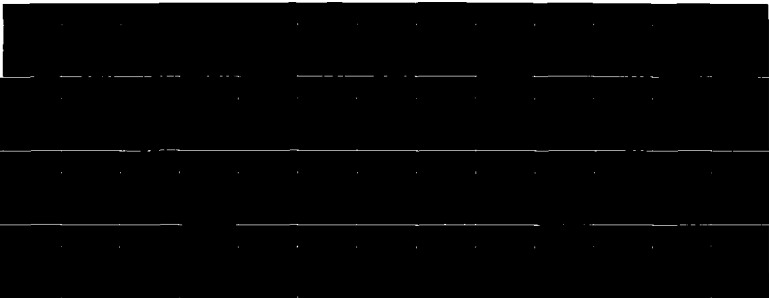
CSS/EMW/SOF (COMBAT SERVICE SUPPORT/ENGINEERING AND
MINE WARFARE/SPECIAL (U) MCLEAN RESEARCH CENTER INC VA
G N ROMSTEDT SEP 86 DAAR70-84-D-0052

4/4

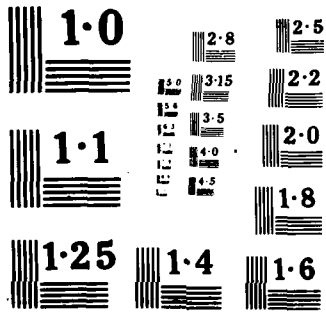
UNCLASSIFIED

F/G 9/2

NL



END
DATE
FILMED
11 86



(Procurement priorities report for type classified systems.
4/23/86 changed to mamp data base.
4/25/86 changed to reflect classification of lrrdp priorities and to add
parameter to deselect printing of those priorities.)

database mamp end

define

variable cnt type integer
variable pno type integer
param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12
param[4] class type character length 1

end

input

prompt for pno using "Please enter the starting page number > "

end

output

left margin 0
right margin 132
report to "procpri.out"

end

read into b

scl_ssn
where scl_code = code and scl_name = name

end

read into a

b
ssn_amc_msc ssn_da_ma
where ssn_acq_code = 2
joining b.scl_ssn = optional ssn_ssn

end

read into c

a
lrrdp_title lrrdp_pdip
lrrdppri_pri
ok = 1
joining a.scl_ssn = lrrdp_ssn
and lrrdp_pdip = lrrdppri_pdip

end

read into b

a
lrrdp_title lrrdp_pdip
lrrdppri_pri
ok = 0
where lrrdppri_pri = 0.0
joining a.scl_ssn = optional lrrdp_ssn
and lrrdp_pdip = optional lrrdppri_pdip

end

Jun 3 14:31 1986 procpri Page 2

```
assign d = c union b end
```

```
sort by ok descending lrrdppri_pri scl_ssn lrrdp_pdip end
```

```
format
```

```
page header
```

```
if class = "U" then print column 47, "***** U N C L A S S I F I E D *****"  
else print column 47, "***** C O N F I D E N T I A L *****"  
skip 2 lines  
print column 47, " DA PROCUREMENT PRIORITY FOR ", name  
skip 2 lines  
print 17 spaces, "SSN", 5 spaces, "DA MA", 4 spaces, "AMC MSC", 15 spaces,  
"TITLE", column 98, " PDIP PRIORITY"  
skip 1 line
```

```
page trailer
```

```
skip 2 lines  
if class = "U" then print column 47, "***** U N C L A S S I F I E D *****"  
else print column 47, "***** C O N F I D E N T I A L *****"  
print column 100, date, 2 spaces, time  
if pletter="Z" then print column 60, pno  
else print column 60, pletter, "-", pno using "###"  
let pno = pno+1
```

```
before group of lrrdp_pdip
```

```
let cnt = cnt+1  
print 10 spaces, cnt using "###. ", scl_ssn, 4 spaces, ssn_da_ma,  
5 spaces, ssn_amc_msc, 2 spaces, lrrdp_title[1, 50], 2 spaces,  
lrrdp_pdip[1, 4], "-", lrrdp_pdip[5, 6], 2 spaces;  
if class = "C" then begin  
if ok=1 then print lrrdppri_pri else print " not assigned"  
end  
else begin  
if ok=1 then print "" else print " not assigned"  
end
```

```
end
```

Jun 4 13:25 1986 projects Page 1

(This report lists the projects in prjctl and checks to see if they are also
in the proj file.)

database mamp end

output
report to "projects.out"
left margin 0
right margin 132
end

read into b
prjctl
proj_title
joining pcl_idx = optional proj_idx
end

sort by pcl_cmd pcl_pe pcl_proj pcl_name end

format

page header
print "COMMAND", 7 spaces, "PE", 5 spaces, "PROJ", 2 spaces, "CAT", 2 spaces,
"SUBCAT", 2 spaces, "CODE", 2 spaces, "NAME", 12 spaces, "TITLE"
skip 2 lines

on every record
print pcl_cmd, 2 spaces, pcl_pe, 2 spaces, pcl_proj, 2 spaces, pcl_cat, 2 spaces,
pcl_subcat, 2 spaces, pcl_code, 2 spaces, pcl_name, 2 spaces, proj_title

end

Jun 3 14:31 1986 projindex Page 1

(Project Index for the proj-wkpkg report.)

database manp end

(last modified 2/6/86 to add mission area parameter and date-time stamp.
3/6/86 to add page letter and page number.
5/12/86 changed to the manp data base.)

```
define
  variable i          type integer
  variable pno        type integer
  param[1] pletter    type character length 1
  param[2] code       type integer
  param[3] name       type character length 12
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  left margin 0
  right margin 132
  report to "projindex.out"
end

read into a
  proj_idx proj_title
  where pcl_code = code and pcl_name = name
  joining pcl_idx = proj_idx
end

sort by proj_pe proj_no proj_cmd end

format

page header
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 3 lines
  print column 51, name clipped, " FUNDED RDTE PROJECT INDEX"
  skip 3 lines
  print 31 spaces, "PE", 5 spaces, "PROJ", 13 spaces, "TITLE",
    28 spaces, "COMMAND", 8 spaces, "PAGE"
  skip 2 lines

before group of proj_no
  let i = i+1

before group of proj_cmd
  print 30 spaces, proj_pe, 3 spaces, proj_no;
  print 3 spaces, proj_title[1, 40], 3 spaces, proj_cmd, 3 spaces, "----"

page trailer
  skip 2 lines
  print column 47, "***** U N C L A S S I F I E D *****"
  print column 100, date, 2 spaces, time
```


Jun 3 14:31 1986 projindex Page 2

```
if pletter="Z" then print 60 spaces,pno  
else print 60 spaces,pletter,"-",pno using "###"  
let pno = pno+1
```

end

{ PROJ - WPKPG summary }

{ last modified 2/6/86 to add mission area parameter and date-time stamp.
2/7/86. added page letter parameter.
Also modified 2/26/86 for Exec. Assmt.
revised 3/3/86 to remove belvoir specifics and to replace peproj in the mamp.
Modified 5/21/86. changed to mamp database.
5/27/86 cleaned up a bit. Added cumulative project funding variables.
6/2/86 removed mandatory check for being in the system control file.
now only reports if in the mission area control file.
6/4/86 added workpackage criticality factor. Sort output by beginning year.
6/6/86 added one extra space for the wpk number. removed one space from
title.
7/23/86 changed wkpsys_srf to wkpsys_srf in keeping with iitri's usage. }

database mamp end

define

variable evalcount type integer
variable pagecount type integer
variable linecount type integer
variable pno type integer
param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12

variable tf0 type long
variable uf0 type long
variable tf1 type long
variable uf1 type long
variable tf2 type long
variable uf2 type long
variable tf3 type long
variable uf3 type long
variable tf4 type long
variable uf4 type long
variable tf5 type long
variable uf5 type long
variable tf6 type long
variable uf6 type long
variable tf7 type long
variable uf7 type long

end

input

prompt for pno using "Please enter the starting page number > "

end

output

right margin 132
left margin 0
report to "projwkp.out"

end

read into d

pcl_idx

```
where pcl_code = code and pcl_name = name
end

read into b
d
proj_idx proj_title
wkpkg_title wkpkg_task wkpkg_no_idx wkpkg_wkpsys_idx wkpkg_pdip wkpfr_baseyr
wkpfr_f0 wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
wkpfr_u0 wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
joining d.pcl_idx = proj_idx
and proj_idx = optional wkpkg_proj_idx
and wkpkg_no_idx = optional wkpfr_wkpkg_idx
end

read into a
b
wkpsys_ssn wkpsys_srf lrrdp_title
joining b.wkpkg_wkpsys_idx = optional wkpsys_wkpkg_idx
and wkpsys_ssn = optional lrrdp_ssn
end

read into e
scl_ssn
where scl_code = code and scl_name = name
end

read into c
a
e
joining a.wkpsys_ssn = optional e.scl_ssn
end

sort by proj_cmd proj_pe proj_no wkpfr_baseyr wkpkg_task wkpkg_no wkpsys_ssn end
format

page header
print column 47, "***** U N C L A S S I F I E D *****"
skip 2 lines
print column 47, " PROJECT/WORKPACKAGE SUMMARY"
skip 2 lines
print 8 spaces, "COMMAND: ", proj_cmd,
8 spaces, "PROJECT: ", proj_pe, 2 spaces, proj_no,
10 spaces, "TITLE: ", proj_title clipped;
if pagecount=1 then print " (Continued)" else print ""
skip 1 line
print " -----",
"-----"
print " : TASK/WKP/SSN", 5 spaces, "WORKPACKAGE/SYSTEM TITLE",
column 67, " : PDIP",
column 73, " : FY86 FY87 FY88 FY89 FY90 FY91 FY92 FY93 !"
print " :-----!-----",
"-----!"
let linecount = 45

page trailer
```

```

skip 1 line
print "Legend", column 47, "***** U N C L A S S I F I E D *****"
print "+ - System designated in Mission Area", column 100, date, 2 spaces, time
if pletter="Z" then print column 60, pno
else print column 60, pletter, "-", pno using "###"
let pno = pno+1

before group of proj_no
let pagecount=0
skip to top of page
let tf0 = 0
let uf0 = 0
let tf1 = 0
let uf1 = 0
let tf2 = 0
let uf2 = 0
let tf3 = 0
let uf3 = 0
let tf4 = 0
let uf4 = 0
let tf5 = 0
let uf5 = 0
let tf6 = 0
let uf6 = 0
let tf7 = 0
let uf7 = 0

after group of proj_no
if linecount < 5 then begin
    print ":", column 67, ":", column 75, ":", column 132, ":"
    print "-----",
    skip to top of page
end
else begin
    print ":", column 67, ":", column 75, ":", column 132, ":"
    print "-----",
end
print ": TOTAL FUNDS FOR PROJECT: FUNDED", column 67, ":", column 75, ":"
if tf0>0 then print tf0 using "##### "; else print " ";
if tf1>0 then print tf1 using "##### "; else print " ";
if tf2>0 then print tf2 using "##### "; else print " ";
if tf3>0 then print tf3 using "##### "; else print " ";
if tf4>0 then print tf4 using "##### "; else print " ";
if tf5>0 then print tf5 using "##### "; else print " ";
if tf6>0 then print tf6 using "##### "; else print " ";
if tf7>0 then print tf7 using "##### "; else print " ";
print column 132, ":"
print ":",
UNFUNDED", column 67, ":", column 75, ":"
if uf0>0 then print -uf0 using "(((((#)"; else print " ";
if uf1>0 then print -uf1 using "(((((#)"; else print " ";
if uf2>0 then print -uf2 using "(((((#)"; else print " ";
if uf3>0 then print -uf3 using "(((((#)"; else print " ";
if uf4>0 then print -uf4 using "(((((#)"; else print " ";
if uf5>0 then print -uf5 using "(((((#)"; else print " ";

```

```

if uf6>0 then print -uf6 using "(((("#); else print " ";
if uf7>0 then print -uf7 using "(((("#); else print " ";
print column 132,"!"
print "-----";

before group of wkpkg_no
let evalcount = 0
if linecount<5 then begin
    print "!",column 67,"!",column 75,"!",column 132,"!"
    print "-----";

    skip to top of page
end
print "!",column 67,"!",column 75,"!",column 132,"!"
print "!",wkpkg_task[1,4],1 space,wkpkg_no,1 space,wkpkg_title[1,49],
    column 67,"!",wkpkg_pdip[1,4],"-",wkpkg_pdip[5,6],column 75,"!";
if wkpfr_f0>0 then print wkpfr_f0 using "##### "; else print " ";
if wkpfr_f1>0 then print wkpfr_f1 using "##### "; else print " ";
if wkpfr_f2>0 then print wkpfr_f2 using "##### "; else print " ";
if wkpfr_f3>0 then print wkpfr_f3 using "##### "; else print " ";
if wkpfr_f4>0 then print wkpfr_f4 using "##### "; else print " ";
if wkpfr_f5>0 then print wkpfr_f5 using "##### "; else print " ";
if wkpfr_f6>0 then print wkpfr_f6 using "##### "; else print " ";
if wkpfr_f7>0 then print wkpfr_f7 using "##### "; else print " ";
let tf0 = tf0 + wkpfr_f0
let tf1 = tf1 + wkpfr_f1
let tf2 = tf2 + wkpfr_f2
let tf3 = tf3 + wkpfr_f3
let tf4 = tf4 + wkpfr_f4
let tf5 = tf5 + wkpfr_f5
let tf6 = tf6 + wkpfr_f6
let tf7 = tf7 + wkpfr_f7
let uf0 = uf0 + wkpfr_u0
let uf1 = uf1 + wkpfr_u1
let uf2 = uf2 + wkpfr_u2
let uf3 = uf3 + wkpfr_u3
let uf4 = uf4 + wkpfr_u4
let uf5 = uf5 + wkpfr_u5
let uf6 = uf6 + wkpfr_u6
let uf7 = uf7 + wkpfr_u7
print "!"
let linecount = linecount-2

after group of wkpkg_no
let pagecount = 1
if evalcount = 0 then begin
    print "!",column 67,"!",column 75,"!";
    if wkpfr_u0>0 then print -wkpfr_u0 using "(((("#); else print " ";
    if wkpfr_u1>0 then print -wkpfr_u1 using "(((("#); else print " ";
    if wkpfr_u2>0 then print -wkpfr_u2 using "(((("#); else print " ";
    if wkpfr_u3>0 then print -wkpfr_u3 using "(((("#); else print " ";
    if wkpfr_u4>0 then print -wkpfr_u4 using "(((("#); else print " ";
    if wkpfr_u5>0 then print -wkpfr_u5 using "(((("#); else print " ";
    if wkpfr_u6>0 then print -wkpfr_u6 using "(((("#); else print " ";
    if wkpfr_u7>0 then print -wkpfr_u7 using "(((("#); else print " ";

```

```
print column 132, ":"
let linecount = linecount-1
end

before group of wkpsys_ssn
if wkpsys_ssn<>" " then begin
if linecount < 3 then begin
print ":", column 67, ":", column 75, ":", column 132, ":"
print "-----",
"-----";

skip to top of page
end
print ":", 5 spaces, wkpsys_ssn, 2 spaces, lrrdp_title[1, 48], 1 space,
wkpsys_srf using "#";
if scl_ssn<>" " then print "*";
print column 67, ":", column 75, ":";
let linecount = linecount-1
if evalcount = 0 then begin
let evalcount = 1
if wkpfr_u0>0 then print -wkpfr_u0 using "(((("#"; else print " ";
if wkpfr_u1>0 then print -wkpfr_u1 using "(((("#"; else print " ";
if wkpfr_u2>0 then print -wkpfr_u2 using "(((("#"; else print " ";
if wkpfr_u3>0 then print -wkpfr_u3 using "(((("#"; else print " ";
if wkpfr_u4>0 then print -wkpfr_u4 using "(((("#"; else print " ";
if wkpfr_u5>0 then print -wkpfr_u5 using "(((("#"; else print " ";
if wkpfr_u6>0 then print -wkpfr_u6 using "(((("#"; else print " ";
if wkpfr_u7>0 then print -wkpfr_u7 using "(((("#"; else print " ";
end
print column 132, ":"
end
end
```

{ PROJ - WKPKG blank summary }

{ last modified 2/6/86 to add mission area parameter and date-time stamp.
2/7/86. added page letter parameter. Also modified 2/26/86 for Exec. Assmt.
revised 3/5/86 to remove belvoir specifics and to replace peproj in the mamp.
5/2/86 moved to the mamp and removed the funding information to produce a
blank table. Changed the planning horizon to fy87 to fy93. }

database mamp end

define

variable evalcount type integer
variable pagecount type integer
variable linecount type integer
variable pno type integer
param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12

end

input

prompt for pno using "Please enter the starting page number > "
end

output

right margin 132
left margin 0
report to "projwkpblk.out"
end

read into a

pcl_idx
where pcl_code = code and pcl_name = name
end

read into b

proj_title
wkpkg_wkpsys_idx wkpkg_title wkpkg_pdip
joining a.pcl_idx = proj_idx
and a.pcl_idx = optional wkpkg_proj_idx
end

read into d

unique scl_ssn lrrdp_title
joining scl_ssn = lrrdp_ssn
end

read into c

b
d
joining b.wkpkg_wkpsys_idx = optional wkpsys_wkpkg_idx
and wkpsys_ssn = optional d.scl_ssn
end

sort by wkpkg_cmd wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no scl_ssn end

format

page header

```
print column 47, "***** U N C L A S S I F I E D *****"
skip 2 lines
print column 47, "      PROJECT/WORKPACKAGE SUMMARY"
skip 2 lines
print 8 spaces, "COMMAND: ", wkpkg_cmd,
      8 spaces, "PROJECT: ", wkpkg_pe, 2 spaces, wkpkg_proj,
      10 spaces, "TITLE: ", proj_title clipped;
if pagecount=1 then print " (Continued)" else print ""
skip 1 line
print "-----",
      "-----"
print " : TASK/WKP/SSN", 5 spaces, "WORKPACKAGE/SYSTEM TITLE",
      column 67, " : PDIP",
      column 75, " :  FY87  FY88  FY89  FY90  FY91  FY92  FY93 :",
print " :-----",
      " :-----";
let linecount = 45
```

page trailer

```
skip 1 line
print column 47, "***** U N C L A S S I F I E D *****"
print column 100, date, 2 spaces, time
if pletter="Z" then print column 60, pno
else print column 60, pletter, "-", pno using "###"
let pno = pno+1
```

before group of wkpkg_proj

```
let pagecount=0
skip to top of page
```

after group of wkpkg_proj

```
if linecount < 5 then begin
  print "!", column 67, "!", column 75, "!", column 132, "!"
  print "-----",
        "-----"
  skip to top of page
end
else begin
  print "!", column 67, "!", column 75, "!", column 132, "!"
  print " :-----",
        " :-----";
end
print " : TOTAL FUNDS FOR PROJECT:  FUNDED", column 67, "!", column 75, "!",
      column 132, "!"
print " :
      UNFUNDED", column 67, "!", column 75, "!",
      column 132, "!"
print " :-----",
      " :-----";
```

before group of wkpkg_no

```
let evalcount = 0
if linecount<5 then begin
  print "!", column 67, "!", column 75, "!", column 132, "!"
```



```
print "-----",
      "-----";
  skip to top of page
end
print "!",column 67,"!",column 75,"!",column 132,"!"
print "!",wkpkg_task[1,4],1 space,wkpkg_no[1,6],2 spaces,wkpkg_title[1,50],
      column 67,"!",wkpkg_pdip[1,4],"-",wkpkg_pdip[5,6],column 75,"!",
      column 132,"!"
let linecount = linecount-2

after group of wkpkg_no
let pagecount = 1
if evalcount = 0 then begin
  print "!",column 67,"!",column 75,"!",column 132,"!"
  let linecount = linecount-1
end

before group of scl_ssn
if scl_ssn<>" " then begin
  if linecount < 3 then begin
    print "!",column 67,"!",column 75,"!",column 132,"!"
    print "-----",
          "-----";
    skip to top of page
  end
  print "!",5 spaces,scl_ssn,2 spaces,lrrdp_title[1,50],
        column 67,"!",column 75,"!";
  let linecount = linecount-1
  if evalcount = 0 then begin
    let evalcount = 1
  end
  print column 132,"!"
end

end
```

Jun 3 14 31 1986 ssndroll Page 1

(this creates the database for the ssndroll file. It is a rollup of the top
8 deficiencies for any one system.)

database manp end

define
variable ndef type integer
end

output
page length 32000
left margin 0
right margin 80
top margin 0
report to "ssndroll.out"
end

read
ssndef_ssn ssndef_def ssndef_cont_value
joining unissn_ssn = ssndef_ssn
end

sort by ssndef_ssn ssndef_def end

format

before group of ssndef_ssn
let ndef = 0
print ssndef_ssn, " !";

before group of ssndef_def
if ndef<8 and ssndef_def>0 then begin
let ndef = ndef+1
print ssndef_def, " !", ssndef_cont_value, " !";
end

after group of ssndef_ssn
let ndef = 8-ndef
if ndef>0 then begin
while ndef>0 do begin
print "0: !";
let ndef = ndef-1
end
end
print group count using "####", " !"

end

Jun 30 16:12 1986 sysdef Page 1

(system to deficiency cross reference check.

6/5/86 taken from priorsys

6/30/86 made unclassified with new guidance.)

database mamp end

define

variable cnt type integer

variable dcnt type integer

variable pno type integer

param[1] pletter type character length 1

param[2] code type integer

param[3] name type character length 12

end

input

prompt for pno using "Please enter the starting page number > "

end

output

left margin 0

right margin 132

report to "sysdef.out"

end

read into b

scl_ssn

where scl_code = code and scl_name = name

end

read into c

b

ssn_amc_mgr ssn_da_ma

lrrdp_title

prior1

joining b.scl_ssn = optional prior1_ssn_no

and b.scl_ssn = optional lrrdp_ssn

and b.scl_ssn = optional ssn_ssn

end

read into a

c

ssndef_def ssndef_cont_value

joining c.scl_ssn = optional ssndef_ssn

end

sort by scl_ssn ssndef_def end

format

page header

print column 47, "***** UNCLASSIFIED *****"

skip 2 lines

print column 44, name clipped, " SYSTEM BDP RANKING AND DEFICIENCIES"

skip 2 lines

Jun 30 16:12 1986 sysdef Page 2

```
print 7 spaces, "SSN", 5 spaces, "DA MA", 4 spaces, "AMC MGR", 15 spaces,
  "TITLE", column 88, " #DEF #A #B #C #D #E #X BL RATING"
skip 1 line

page trailer
skip 2 lines
print column 47, "***** U N C L A S S I F I E D *****"
print column 100, date, 2 spaces, time
if pletter="Z" then print column 60, pno
else print column 60, pletter, "-", pno using "###"
let pno = pno+1

before group of scl_ssn
let cnt = cnt+1
print cnt using "####. ", scl_ssn, 4 spaces, ssn_da_ma,
  5 spaces, ssn_amc_mgr, column 30, lrrdp_title[1, 50], 2 spaces,
  priori_ndef using "####", priori_na using "####", priori_nb using "####",
  priori_nc using "####", priori_nd using "####",
  priori_ne using "####", priori_nx using "####", priori_bl using "####",
  priori_score using "#####. ##"
let dcnt = 0
print 15 spaces;

after group of scl_ssn
print ""

before group of ssndef_def
let dcnt = dcnt+1
if dcnt>15 then begin
  let dcnt=1
  print ""
  print 15 spaces;
end
print ssndef_def using " ####", "-", ssndef_cont_value;

end
```

Aug 4 09:30 1986 sysdef2 Page 1

(abbreviated system to deficiency cross reference check.
removed system titles and prior data from sysdef.

8/4/86. taken from sysdef.)

database mamp end

define

variable dcnt type integer
variable pno type integer
param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12
param[4] cmd type character length 12

end

input

prompt for pno using "Please enter the starting page number > "
end

output

left margin 0
right margin 132
report to "sysdef2.out"
end

read into b

scl_ssn
where scl_code = code and scl_name matches name
end

read into c

b
where ssn_amc_mgr matches cmd
joining b.scl_ssn = ssn_ssn
end

read into a

c
ssndef_def ssndef_cont_value
joining c.scl_ssn = optional ssndef_ssn
end

sort by scl_ssn ssndef_def end

format

page header

print column 47, "***** U N C L A S S I F I E D *****"
skip 2 lines
print column 50, name clipped, " SYSTEM BDP DEFICIENCIES"
skip 2 lines
print 2 spaces, "SSN", 5 spaces, "DEFICIENCIES"
skip 1 line

page trailer

Aug 4 09:30 1986 sysdef2 Page 2

```
skip 2 lines
print column 47, "***** U N C L A S S I F I E D *****"
print column 100, date, 2 spaces, time
if pletter="Z" then print column 60, pno
else print column 60, pletter, "-", pno using "###"
let pno = pno+1

before group of scl_ssn
print scl_ssn, 4 spaces;
let dcnt = 0

after group of scl_ssn
print ""

before group of ssndef_def
let dcnt = dcnt+1
if dcnt>15 then begin
  let dcnt=1
  print ""
  print 10 spaces;
end
print ssndef_def using " ###", "-", ssndef_cont_value;

end
```

(System Summary with Dollar Values inserted.

Last modified on 2/4/86 to add date-time stamp, tradoc ma,
ma parameter, and workpackage rating factor.
2/7/86 to change commodity and mission area designators.
5/12/86 changed to the mamp database. changed the logic so that all systems
with any workpackages and all developmental systems - regardless of
workpackages - will print. Added page letter parameter.
5/13/86 removed *1000 in lrrdp funded amounts. It's already in \$1000s.
5/16/86 added report group parameter (t)
5/21/86 changed parameter t to "matches" condition.
6/9/86 removed group parameter (t), this report will print all systems
including type classified. Removed < completed >.
6/13/86 changed classification to include the possibility of a secret
system description. Get classification from ssn_class.
7/3/86 corrected project linkage to allow workpackages in control file
linked to other mission areas.
7/10/86 changed wkpsys_wc to wkpsys_srf in keeping with iitri's usage.
7/15/86 extensive modifications to print multiple pdips. }

database mamp end

```
define
  variable pfmt          type character length 8
  variable nfmt          type character length 8
  variable bfmt          type character length 8
  variable proghead     type character length 30
  variable vb           type character length 1
  variable head         type integer
  variable phead        type integer
  variable unf          type integer
  variable hflag        type integer
  variable tbflag       type integer
  variable printflag    type integer
  variable break        type integer
  variable obreak       type integer
  variable pagebreak    type integer
  variable linesleft    type integer
  variable wtot         type integer
  variable evalcount    type integer
  variable pno          type integer
  param[1] pletter      type character length 1
  param[2] code         type integer
  param[3] name         type character length 12
  param[4] cmd          type character length 12
  param[5] ssn         type character length 6
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  left margin 0
  right margin 132
  report to "sysdollars.out"
```

```
end

read into c
  scl_ssn
  where scl_code = code and scl_name = name and scl_ssn matches ssn
end

read into a
  c
  acq_code acq_type
  ssn_desc ssn_req_doc1 ssn_req_doc2 ssn_req_doc3 ssn_amc_msc ssn_com_line
  ssn_ifuncarea ssn_amc_mgr ssn_tradoc_pro ssn_tradoc_ma ssn_da_ma ssn_class
  where ssn_amc_mgr matches cmd
  joining c.scl_ssn = optional ssn_ssn
         and ssn_acq_code = optional acq_code
end

read into b
  a
  lrrdp_pdip lrrdp_title
  lrrdp_procf1 lrrdp_procf2 lrrdp_procf3 lrrdp_procf4 lrrdp_procf5
  lrrdp_procf6 lrrdp_procf7 lrrdp_procf8 lrrdp_procf9 lrrdp_procf10
  lrrdp_procf11 lrrdp_procf12 lrrdp_procf13 lrrdp_procf14
  lrrdppri_pri
  ssndroll
  asst
  joining a.scl_ssn = optional lrrdp_ssn
         and lrrdp_pdip = optional lrrdppri_pdip
         and a.scl_ssn = optional ssnd_ssn
         and a.scl_ssn = optional asst_ssn
end

read into c
  prjct1
  where pcl_code = code and pcl_name = name
end

read into a
  b
  wkpsys_srf
  prior2_nsys
  wkpkg_cmd wkpkg_cat wkpkg_subcat wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no
  wkpkg_title wkpfr_baseyr
  wkpfr_f0 wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
  wkpfr_u0 wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
  c.pcl_name
  joining b.scl_ssn = optional wkpsys_ssn
         and wkpsys_wkpkg_idx = optional wkpkg_wkpsys_idx
         and wkpkg_no_idx = optional wkpfr_wkpkg_idx
         and wkpkg_proj_idx = optional c.pcl_idx
         and wkpkg_no_idx = optional prior2_wkpkg_idx
end

sort by scl_ssn wkpkg_subcat descending
  wkpsys_srf wkpkg_pe wkpkg_proj wkpkg_no lrrdp_pdip end
```



```

format

page header
  if ssn_class = "S" then print column 52, "***** S E C R E T *****"
  else print column 47, "***** C O N F I D E N T I A L *****"
  skip 2 lines
  print 20 spaces, "SSN: ", scl_ssn, 20 spaces, "TITLE: ", lrrdp_title
  let pagebreak = 0
  let printflag = 1
  let linesleft = 51
  let vb = "!"
  let pfmt = "##### "
  let nfmt = "((((((#)"
  let bfmt = " "

page trailer
  skip 1 line
  print "Legend: ";
  if ssn_class = "S" then print column 52, "***** S E C R E T *****"
  else print column 47, "***** C O N F I D E N T I A L *****"
  print " v - Workpackage Contribution Value", column 100, date, 2 spaces, time
  print " * - ", name clipped, " Funded Workpackage"
  print " m - Workpackage Supports Multiple Systems";
  if pletter = "Z" then print column 60, pno
  else print column 60, pletter, "-", pno using "###"
  let pno = pno+1

before group of scl_ssn
  skip to top of page
  let tbflag = 0
  let head = 1

  if asrr_ssn1<>" " then begin
    skip 1 line
    print 20 spaces, "ASSOC SSN: ";
    if asrr_ssn1<>" " then print 4 spaces, asrr_ssn1;
    if asrr_ssn2<>" " then print 4 spaces, asrr_ssn2;
    if asrr_ssn3<>" " then print 4 spaces, asrr_ssn3;
    if asrr_ssn4<>" " then print 4 spaces, asrr_ssn4;
    if asrr_ssn5<>" " then print 4 spaces, asrr_ssn5;
    if asrr_ssn6<>" " then print 4 spaces, asrr_ssn6;
    if asrr_ssn7<>" " then print 4 spaces, asrr_ssn7;
    if asrr_ssn8<>" " then print 4 spaces, asrr_ssn8;
    if asrr_ssn9<>" " then print 4 spaces, asrr_ssn9;
    if asrr_ssn10<>" " then print 4 spaces, asrr_ssn10;
    print " "
    let linesleft = linesleft-2
  end

  skip 1 line
  print "DA/TRADOC Mission Area: ", ssn_da_ma, " / ", ssn_tradoc_ma,
    column 80, "AMCMSC: ", ssn_amc_msc
  print "Commodity Line: ", ssn_com_line,
    column 80, "AMC Manager: ", ssn_amc_mgr
  print "Cross Functional Area: ", ssn_xfuncarea,
    column 80, "TRADOC Proponent: ", ssn_tradoc_pro

```

```

print "Acquisition Type: ",acq_type,
      column 80,"Req. Document: ",("ssn_req_doc1,
      ",",ssn_req_doc2,"",",ssn_req_doc3,")"
let linesleft = linesleft-5

skip 1 line
print "TOP 8 DEFICIENCIES: ";
if ssnd_def1>0 then begin
  print ssnd_def1;
  if ssnd_con1<>" " then print "-",ssnd_con1; else print " "; end
if ssnd_def2>0 then begin
  print ", ",ssnd_def2;
  if ssnd_con2<>" " then print "-",ssnd_con2; else print " "; end
if ssnd_def3>0 then begin
  print ", ",ssnd_def3;
  if ssnd_con3<>" " then print "-",ssnd_con3; else print " "; end
if ssnd_def4>0 then begin
  print ", ",ssnd_def4;
  if ssnd_con4<>" " then print "-",ssnd_con4; else print " "; end
if ssnd_def5>0 then begin
  print ", ",ssnd_def5;
  if ssnd_con5<>" " then print "-",ssnd_con5; else print " "; end
if ssnd_def6>0 then begin
  print ", ",ssnd_def6;
  if ssnd_con6<>" " then print "-",ssnd_con6; else print " "; end
if ssnd_def7>0 then begin
  print ", ",ssnd_def7;
  if ssnd_con7<>" " then print "-",ssnd_con7; else print " "; end
if ssnd_def8>0 then begin
  print ", ",ssnd_def8;
  if ssnd_con8<>" " then print "-",ssnd_con8; else print " "; end
print " "
let linesleft = linesleft -2

skip 1 line
print "DESCRIPTION: ";
let break = 118
while ssn_desc[break,break]<>" " do let break = break-1
print ssn_desc[1,break]
let obreak = break+1
let break = break+118
while ssn_desc[break,break]<>" " do let break = break-1
print 14 spaces,ssn_desc[obreak,break]
let obreak = break+1
let break = break+118
while ssn_desc[break,break]<>" " do let break = break-1
print 14 spaces,ssn_desc[obreak,break]
let obreak = break+1
let break = break+118
while ssn_desc[break,break]<>" " do let break = break-1
print 14 spaces,ssn_desc[obreak,break]
let obreak = break+1

```

```

print 14 spaces, ssn_desc[obreak, 600]
skip 2 lines
let linesleft = linesleft-9

before group of lrrdp_pdip

if head = 1 then begin
print column 56, "PRODUCTION PROGRAM ($K)"
print "-----"
print ": PDIP/INCR",
column 18, ": 94 95 96 97 98 99 :",
" 86 87 88 89 90 91 92 93 :",
print "-----";
let head = 0
let phead = 1
let linesleft = linesleft-4
end

if phead = 1 then begin
if lrrdp_pri_pri > 800.0 or lrrdp_pri_pri < 1.0 then let unf=1 else let unf=0
print vb, 4 spaces, lrrdp_pdip[1, 4], "-", lrrdp_pdip[5, 6], column 18, vb;
if unf=0 then begin
if lrrdp_proc9 > 0 then print lrrdp_proc9 using pfmt; else print bfmt;
if lrrdp_proc10 > 0 then print lrrdp_proc10 using pfmt; else print bfmt;
if lrrdp_proc11 > 0 then print lrrdp_proc11 using pfmt; else print bfmt;
if lrrdp_proc12 > 0 then print lrrdp_proc12 using pfmt; else print bfmt;
if lrrdp_proc13 > 0 then print lrrdp_proc13 using pfmt; else print bfmt;
if lrrdp_proc14 > 0 then print lrrdp_proc14 using pfmt; else print bfmt;
print vb;
if lrrdp_proc1 > 0 then print lrrdp_proc1 using pfmt; else print bfmt;
if lrrdp_proc2 > 0 then print lrrdp_proc2 using pfmt; else print bfmt;
if lrrdp_proc3 > 0 then print lrrdp_proc3 using pfmt; else print bfmt;
if lrrdp_proc4 > 0 then print lrrdp_proc4 using pfmt; else print bfmt;
if lrrdp_proc5 > 0 then print lrrdp_proc5 using pfmt; else print bfmt;
if lrrdp_proc6 > 0 then print lrrdp_proc6 using pfmt; else print bfmt;
if lrrdp_proc7 > 0 then print lrrdp_proc7 using pfmt; else print bfmt;
if lrrdp_proc8 > 0 then print lrrdp_proc8 using pfmt; else print bfmt;
end
else begin
if lrrdp_proc9 > 0 then print -lrrdp_proc9 using nfmt; else print bfmt;
if lrrdp_proc10 > 0 then print -lrrdp_proc10 using nfmt; else print bfmt;
if lrrdp_proc11 > 0 then print -lrrdp_proc11 using nfmt; else print bfmt;
if lrrdp_proc12 > 0 then print -lrrdp_proc12 using nfmt; else print bfmt;
if lrrdp_proc13 > 0 then print -lrrdp_proc13 using nfmt; else print bfmt;
if lrrdp_proc14 > 0 then print -lrrdp_proc14 using nfmt; else print bfmt;
print vb;
if lrrdp_proc1 > 0 then print -lrrdp_proc1 using nfmt; else print bfmt;
if lrrdp_proc2 > 0 then print -lrrdp_proc2 using nfmt; else print bfmt;
if lrrdp_proc3 > 0 then print -lrrdp_proc3 using nfmt; else print bfmt;
if lrrdp_proc4 > 0 then print -lrrdp_proc4 using nfmt; else print bfmt;
if lrrdp_proc5 > 0 then print -lrrdp_proc5 using nfmt; else print bfmt;
if lrrdp_proc6 > 0 then print -lrrdp_proc6 using nfmt; else print bfmt;
if lrrdp_proc7 > 0 then print -lrrdp_proc7 using nfmt; else print bfmt;
if lrrdp_proc8 > 0 then print -lrrdp_proc8 using nfmt; else print bfmt;
end
end

```

```
end
print column 132,vb
let linesleft = linesleft -1
end

before group of wkpkg_subcat
if proghead = " " then let printflag = 1
if (wkpkg_subcat="6.1" or wkpkg_subcat="6.2" or wkpkg_subcat="6.3A")
then begin
if tbflag = 1 then begin
let printflag = 1
let tbflag = 0
print "-----",
"-----"
let linesleft = linesleft-1
if linesleft < 9 then skip to top of page
end
let proghead = "TECH BASE PROGRAMS ($K)"
end
else begin
let tbflag = 1
let proghead = "DEVELOPMENT PROGRAMS ($K)"
end
end

after group of wkpkg_no

if phead = 1 then begin
print "-----",
"-----"
skip 2 lines
let linesleft = linesleft-3
let phead = 0
end

if linesleft < 3 then begin
print "-----",
"-----"
skip to top of page
skip 2 lines
let linesleft = linesleft-2
end

if printflag = 1 then begin
skip 2 lines
print column 55,proghead
print "-----",
"-----"
print vb,column 63,"|v*m|",
"      86      87      88      89      90      91      92      93 :|"
print "|:-----|",
"|:-----|:"
let linesleft = linesleft-6
let printflag = 0
end

let break = 32
```

```

if wkpkg_pe=" " then print ": < No identified workpackages >";
else begin
  print ": ",wkpkg_pe,1 space,wkpkg_proj,1 space,wkpkg_task[1,3],1 space,
    wkpkg_cmd[1,4],1 space,wkpkg_no,1 space;
  while wkpkg_title[break,break]<>" " do let break = break-1
  print wkpkg_title[1, break];
end
print column 63,vb,wkpsys_srf using "#";
if pcl_name = name then print "*"; else print " ";
if prior2_nsys>1 then print "m",vb; else print " ",vb;
let wtot = wkpfr_f0+wkpfr_f1+wkpfr_f2+wkpfr_f3+wkpfr_f4+wkpfr_f5+
  wkpfr_f6+wkpfr_f7+wkpfr_u0+wkpfr_u1+wkpfr_u2+wkpfr_u3+
  wkpfr_u4+wkpfr_u5+wkpfr_u6+wkpfr_u7
if wtot > 0 then begin
  if wkpfr_f0>0 then print wkpfr_f0 using pfmt; else print bfmt;
  if wkpfr_f1>0 then print wkpfr_f1 using pfmt; else print bfmt;
  if wkpfr_f2>0 then print wkpfr_f2 using pfmt; else print bfmt;
  if wkpfr_f3>0 then print wkpfr_f3 using pfmt; else print bfmt;
  if wkpfr_f4>0 then print wkpfr_f4 using pfmt; else print bfmt;
  if wkpfr_f5>0 then print wkpfr_f5 using pfmt; else print bfmt;
  if wkpfr_f6>0 then print wkpfr_f6 using pfmt; else print bfmt;
  if wkpfr_f7>0 then print wkpfr_f7 using pfmt; else print bfmt;
end
else if wkpfr_baseyr>1993 then print column 100,"<scheduled out year>";
else print " <not scheduled>";
print column 132,vb
if break > 29 then print vb,column 32,wkpkg_title[break+1,60];
else print vb,column 32,wkpkg_title[break+1, break+31];
print column 63,vb," ",vb;
if wkpfr_u0>0 then print -wkpfr_u0 using nfmt; else print bfmt;
if wkpfr_u1>0 then print -wkpfr_u1 using nfmt; else print bfmt;
if wkpfr_u2>0 then print -wkpfr_u2 using nfmt; else print bfmt;
if wkpfr_u3>0 then print -wkpfr_u3 using nfmt; else print bfmt;
if wkpfr_u4>0 then print -wkpfr_u4 using nfmt; else print bfmt;
if wkpfr_u5>0 then print -wkpfr_u5 using nfmt; else print bfmt;
if wkpfr_u6>0 then print -wkpfr_u6 using nfmt; else print bfmt;
if wkpfr_u7>0 then print -wkpfr_u7 using nfmt; else print bfmt;
print column 132,vb
let linesleft = linesleft-2

after group of scl_ssn
print " -----",
"-----";

end

```

database mamp end

{ last modified 2/6/86 to add date-time stamp and mission area.
2/7/86 to get title and mission areas from ssndesc.
3/5/86 to add pdip and proponent entries
3/6/86 to add page letter and number
4/23/86 switch to the mamp data base.
4/24/86 added acquisition code.
5/2/86 only the highest priority pdip is now printed.
6/13/86 added idxpg_no for page numbers.
7/11/86 changed "command" to "amc mgr".}

```
define
  param[1] pletter type character length 1
  param[2] code     type integer
  param[3] name     type character length 12
  variable pno     type integer
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  left margin 0
  right margin 132
  report to "sysindex.out"
end

read into b
  scl_ssn
  where scl_code = code and scl_name = name
end

read into a
  b
  ssn_amc_mgr ssn_da_ma ssn_tradoc_ma ssn_tradoc_pro
  hipri_pdip hipri_title acq_stt
  idxpg_no
  joining b.scl_ssn = optional ssn_ssn
         and b.scl_ssn = optional hipri_ssn
         and b.scl_ssn = optional idxpg_ssn
         and ssn_acq_code = optional acq_code
end

sort by hipri_title scl_ssn idxpg_no end

format

before group of scl_ssn
  print 6 spaces, scl_ssn, 4 spaces, ssn_da_ma,
        5 spaces, ssn_tradoc_ma, " ", ssn_tradoc_pro, 4 spaces,
        hipri_title[1, 50], 4 spaces, ssn_amc_mgr, 4 spaces, acq_stt,
        4 spaces, hipri_pdip[1, 4], "-", hipri_pdip[5, 6], 5 spaces;
  if idxpg_no > 0 then print idxpg_no using "####"
  else print ""
```

```
page header
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 45, name clipped, " SYSTEMS IN ALPHABETICAL ORDER BY TITLE"
  skip 2 lines
  print 7 spaces, "SSN", 5 spaces, "DA MA", 3 spaces, "TDC MA/PROP",
    18 spaces, "SYSTEM TITLE",
    27 spaces, "AMC MGR", 9 spaces, "ACG", 6 spaces, "PDIP", 7 spaces, "PAGE"
  skip 2 lines

page trailer
  skip 2 lines
  print column 47, "***** U N C L A S S I F I E D *****"
  print column 100, date, 2 spaces, time
  if pletter="Z" then print 62 spaces, pno
  else print 62 spaces, pletter, "-", pno using "###"
  let pno = pno+1

end
```

database mamp end

```
( last modified 2/6/86 to add date-time stamp and mission area.  
  2/7/86 to get title and mission areas from ssndesc.  
  3/3/86 to add pdip and proponent entries  
  3/6/86 to add page letter and number  
  4/23/86 switch to the mamp data base.  
  4/24/86 added acquisition code.  
  5/2/86 changed to highest priority pdip  
  6/13/86 added idxpg_no for page number.  
  7/11/86 changed "command" to "amc mgr". )
```

```
define  
  param[1] pletter type character length 1  
  param[2] code     type integer  
  param[3] name     type character length 12  
  variable pno      type integer  
end  
  
input  
  prompt for pno using "Please enter the starting page number > "  
end  
  
output  
  left margin 0  
  right margin 132  
  report to "sysindex1.out"  
end  
  
read into b  
  scl_ssn  
  where scl_code = code and scl_name = name  
end  
  
read into a  
  b  
  ssn_amc_mgr ssn_da_ma ssn_tradoc_ma ssn_tradoc_pro  
  hipri_pdip hipri_title acq_stt  
  idxpg_no  
  joining b.scl_ssn = optional ssn_ssn  
  and b.scl_ssn = optional hipri_ssn  
  and b.scl_ssn = optional idxpg_ssn  
  and ssn_acq_code = optional acq_code  
end  
  
sort by scl_ssn idxpg_no end  
  
format  
  
before group of scl_ssn  
  print 6 spaces, scl_ssn, 4 spaces, ssn_da_ma,  
  5 spaces, ssn_tradoc_ma, " ", ssn_tradoc_pro 4 spaces,  
  hipri_title[1,30], 4 spaces, ssn_amc_mgr, 4 spaces, acq_stt,  
  4 spaces, hipri_pdip[1,4], "-", hipri_pdip[5,6], 4 spaces;  
  if idxpg_no>0 then print idxpg_no using "####"  
  else print ""
```


Jul 11 14:28 1986 sysindex1 Page 2

```
page header
  print column 47, "***** UNCLASSIFIED *****"
  skip 2 lines
  print column 48, name clipped, " SYSTEMS IN NUMERICAL ORDER BY SSN"
  skip 2 lines
  print 7 spaces, "SSN", 5 spaces, "DA MA", 3 spaces, "TDC MA/PROP",
    18 spaces, "SYSTEM TITLE",
    27 spaces, "AMC MGR", 9 spaces, "ACQ", 6 spaces, "PDIP", 7 spaces, "PAGE"
  skip 2 lines

page trailer
  skip 2 lines
  print column 47, "***** UNCLASSIFIED *****"
  print column 100, date, 2 spaces, time
  if pletter="Z" then print 62 spaces, pno
  else print 62 spaces, pletter, "-", pno using "###"
  let pno = pno+1

end
```

< SYSTEM RTE FUNDING ROLLUP PRODUCER. This program is the precursor to defroll. It summarizes the rte funding for a system. It does not include 6.5 funding for a system. >

- < 5/19/86. changed to the mamp database.
- 5/20/86. made corrections.
- 6/10/86. removed the code and name parameters. This program now works for all systems in the wkpsys file and should be updated occasionally from the shell syspro1.
- 7/23/86. changed from year to generic data for the fields.
- 7/25/86. changed to the file structure of fundpro. reinstated the code and name parameters.>

database mamp end

```
define
  param[1]   code           type integer
  param[2]   name           type character length 12
  variable   v              type character length 1
  variable   f0_tb          type long
  variable   u0_tb          type long
  variable   f0_dev         type long
  variable   u0_dev         type long
  variable   f1_tb          type long
  variable   u1_tb          type long
  variable   f1_dev         type long
  variable   u1_dev         type long
  variable   f2_tb          type long
  variable   u2_tb          type long
  variable   f2_dev         type long
  variable   u2_dev         type long
  variable   f3_tb          type long
  variable   u3_tb          type long
  variable   f3_dev         type long
  variable   u3_dev         type long
  variable   f4_tb          type long
  variable   u4_tb          type long
  variable   f4_dev         type long
  variable   u4_dev         type long
  variable   f5_tb          type long
  variable   u5_tb          type long
  variable   f5_dev         type long
  variable   u5_dev         type long
  variable   f6_tb          type long
  variable   u6_tb          type long
  variable   f6_dev         type long
  variable   u6_dev         type long
  variable   f7_tb          type long
  variable   u7_tb          type long
  variable   f7_dev         type long
  variable   u7_dev         type long
end
```

```
output
  top margin 0
  bottom margin 0
```

Jul 25 13:02 1986 syspro1 Page 2

```
left margin 0
page length 32000
report to "syspro1.out"
end

read into b
dcl_def
where dcl_code = code and dcl_name = name
end

read into c
unique ssndef_ssn
joining b.dcl_def = ssndef_def
end

read into a
wkpsys_cmd wkpsys_cat wkpsys_subcat wkpsys_wkpkg wkpsys_ssn
wkpfr_f0 wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
wkpfr_u0 wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
joining c.ssndef_ssn = wkpsys_ssn
and wkpsys_wkpkg_idx = wkpkg_wkpsys_idx
and wkpkg_no_idx = wkpfr_wkpkg_idx
end

sort by wkpsys_ssn wkpsys_cmd wkpsys_cat wkpsys_wkpkg end

format

before group of wkpsys_ssn
let f0_tb = 0
let u0_tb = 0
let f0_dev = 0
let u0_dev = 0
let f1_tb = 0
let u1_tb = 0
let f1_dev = 0
let u1_dev = 0
let f2_tb = 0
let u2_tb = 0
let f2_dev = 0
let u2_dev = 0
let f3_tb = 0
let u3_tb = 0
let f3_dev = 0
let u3_dev = 0
let f4_tb = 0
let u4_tb = 0
let f4_dev = 0
let u4_dev = 0
let f5_tb = 0
let u5_tb = 0
let f5_dev = 0
let u5_dev = 0
let f6_tb = 0
let u6_tb = 0
let f6_dev = 0
```

```
let u6_dev = 0
let f7_tb = 0
let u7_tb = 0
let f7_dev = 0
let u7_dev = 0
```

before group of wkpsys_wkpkg

```
if wkpsys_subcat="6.1" or wkpsys_subcat="6.2" or wkpsys_subcat="6.3A" then begin
  let f0_tb = f0_tb + wkpfr_f0
  let f1_tb = f1_tb + wkpfr_f1
  let f2_tb = f2_tb + wkpfr_f2
  let f3_tb = f3_tb + wkpfr_f3
  let f4_tb = f4_tb + wkpfr_f4
  let f5_tb = f5_tb + wkpfr_f5
  let f6_tb = f6_tb + wkpfr_f6
  let f7_tb = f7_tb + wkpfr_f7
  let u0_tb = u0_tb + wkpfr_u0
  let u1_tb = u1_tb + wkpfr_u1
  let u2_tb = u2_tb + wkpfr_u2
  let u3_tb = u3_tb + wkpfr_u3
  let u4_tb = u4_tb + wkpfr_u4
  let u5_tb = u5_tb + wkpfr_u5
  let u6_tb = u6_tb + wkpfr_u6
  let u7_tb = u7_tb + wkpfr_u7
end
```

```
else if wkpsys_subcat="6.3B" or wkpsys_subcat="6.4" or wkpsys_subcat="6.7" then begin
```

```
  let f0_dev = f0_dev + wkpfr_f0
  let f1_dev = f1_dev + wkpfr_f1
  let f2_dev = f2_dev + wkpfr_f2
  let f3_dev = f3_dev + wkpfr_f3
  let f4_dev = f4_dev + wkpfr_f4
  let f5_dev = f5_dev + wkpfr_f5
  let f6_dev = f6_dev + wkpfr_f6
  let f7_dev = f7_dev + wkpfr_f7
  let u0_dev = u0_dev + wkpfr_u0
  let u1_dev = u1_dev + wkpfr_u1
  let u2_dev = u2_dev + wkpfr_u2
  let u3_dev = u3_dev + wkpfr_u3
  let u4_dev = u4_dev + wkpfr_u4
  let u5_dev = u5_dev + wkpfr_u5
  let u6_dev = u6_dev + wkpfr_u6
  let u7_dev = u7_dev + wkpfr_u7
end
```

after group of wkpsys_ssn

```
let v="!"
print "0", v, " ", v, wkpsys_ssn, v;
print f0_tb, v, f1_tb, v, f2_tb, v, f3_tb, v, f4_tb, v, f5_tb, v, f6_tb, v, f7_tb, v;
print u0_tb, v, u1_tb, v, u2_tb, v, u3_tb, v, u4_tb, v, u5_tb, v, u6_tb, v, u7_tb, v;
print f0_dev, v, f1_dev, v, f2_dev, v, f3_dev, v, f4_dev, v, f5_dev, v, f6_dev, v, f7_dev, v;
print u0_dev, v, u1_dev, v, u2_dev, v, u3_dev, v, u4_dev, v, u5_dev, v, u6_dev, v, u7_dev, v
```

end

(System Resource Report with RDTE Funding Chart.

Modified on 2/5/86 to include mission area parameter and date-time stamp on each page.

Modified on 5/19/86 to add command and group parameters.

and to change report in order to run in mamp.

Made corrections. 5/20/86

5/29/86 merged into a single file for all commands as an option.

6/10/86 removed the group parameter.)

database mamp end

define

variable wkp type integer

variable tbo type integer

variable pgend type integer

variable f85 type float

variable f86 type float

variable f87 type float

variable f88 type float

variable f89 type float

variable f90 type float

variable f91 type float

variable f92 type float

variable u85 type float

variable u86 type float

variable u87 type float

variable u88 type float

variable u89 type float

variable u90 type float

variable u91 type float

variable u92 type float

variable pno type integer

param[1] pletter type character length 1

param[2] code type integer

param[3] name type character length 12

param[4] cmd type character length 12

end

input

prompt for pno using "Please enter the starting page number > "

end

output

left margin 0

right margin 132

report to "sysres.out"

end

read into c

scl_ssn scl_code

where scl_code = code and scl_name = name

end

read into a

c

```
hipri_title
ssn_anc_mgr ssn_da_ma ssn_tradoc_ma
where ssn_anc_mgr matches cmd
and (ssn_acq_code = 3 or ssn_acq_code = 4)
(excludes base case, type classified, tech demo, and BBTA systems.)
joining c.scl_ssn = ssn_ssn
and c.scl_ssn = hipri_ssn
end

read into c
a
wkpkg_no_idx wkpkg_subcat
wkpfr_f0 wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
wkpfr_u0 wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
joining a.scl_ssn = optional wkpsys_ssn
and wkpsys_wkpkg_idx = optional wkpkg_wkpsys_idx
and wkpkg_no_idx = wkpfr_wkpkg_idx
end

sort by scl_ssn end

format

page header
print column 47, "***** U N C L A S S I F I E D *****"
skip 2 lines
print column 42, "SYSTEMS WITHOUT ACTIVE 6.3B/6.4 WORKPACKAGES FOR ", name
skip 3 lines
print column 90, "DEVELOPMENT SCHEDULE"
print "-----"
print " : SSN", 10 spaces, "TITLE", column 57, "DA/TDC MA", column 67,
" : FY86 FY87 FY88 FY89 FY90 FY91 FY92 FY93 : "
print "-----"
print " : ", column 67, " : ", column 132, " : "
let pgend = 0

page trailer
if pgend = 0 then begin
print " : ", column 67, " : ", column 132, " : "
print "-----"
end
else skip 2 lines
skip 2 lines
print "Legend:", column 47, "***** U N C L A S S I F I E D *****"
print "***** Funded (Tech Base Only)", column 100, date, 2 spaces, time
print "←-←-←- Partial Funded"
print "----- Unfunded";
if pletter = "Z" then print column 60, pno
else print column 60, pletter, "-", pno using "###"
let pno = pno+1

on last record
let pgend = 1
```

```
print ":", column 67, ":", column 132, "!"
print "-----",
      "-----"
```

before group of scl_ssn

```
let wkp = 1
let tbo = 1
let f85 = 0
let f86 = 0
let f87 = 0
let f88 = 0
let f89 = 0
let f90 = 0
let f91 = 0
let f92 = 0
let u85 = 0
let u86 = 0
let u87 = 0
let u88 = 0
let u89 = 0
let u90 = 0
let u91 = 0
let u92 = 0
```

after group of scl_ssn

```
if wkp=1 or tbo=1 then begin
  print ":", scl_ssn, 2 spaces, hipri_title(1, 45), 1 space,
        ssn_da_ma, " / ", ssn_tradoc_ma, column 67, "!"
  if f85>0 then begin
    if u85>0 then print "***---"; else print "*****"; end
    else if u85>0 then print "-----"; else print " ";
  if f86>0 then begin
    if u86>0 then print "***---"; else print "*****"; end
    else if u86>0 then print "-----"; else print " ";
  if f87>0 then begin
    if u87>0 then print "***---"; else print "*****"; end
    else if u87>0 then print "-----"; else print " ";
  if f88>0 then begin
    if u88>0 then print "***---"; else print "*****"; end
    else if u88>0 then print "-----"; else print " ";
  if f89>0 then begin
    if u89>0 then print "***---"; else print "*****"; end
    else if u89>0 then print "-----"; else print " ";
  if f90>0 then begin
    if u90>0 then print "***---"; else print "*****"; end
    else if u90>0 then print "-----"; else print " ";
  if f91>0 then begin
    if u91>0 then print "***---"; else print "*****"; end
    else if u91>0 then print "-----"; else print " ";
  if f92>0 then begin
    if u92>0 then print "***---"; else print "*****"; end
    else if u92>0 then print "-----"; else print " ";
  print column 132, "!"
end
```

on every record

```
if wkpkg_no<>" " then let wkp = 0
if wkpkg_subcat="6.38" or wkpkg_subcat="6.4" or wkpkg_subcat="6.5"
or wkpkg_subcat="6.7" then let tbo=0
let f85 = f85 + wkpfr_f0
let u85 = u85 + wkpfr_u0
let f86 = f86 + wkpfr_f1
let u86 = u86 + wkpfr_u1
let f87 = f87 + wkpfr_f2
let u87 = u87 + wkpfr_u2
let f88 = f88 + wkpfr_f3
let u88 = u88 + wkpfr_u3
let f89 = f89 + wkpfr_f4
let u89 = u89 + wkpfr_u4
let f90 = f90 + wkpfr_f5
let u90 = u90 + wkpfr_u5
let f91 = f91 + wkpfr_f6
let u91 = u91 + wkpfr_u6
let f92 = f92 + wkpfr_f7
let u92 = u92 + wkpfr_u7
```

end


```
database mamp end

define
  variable hissn type character length 6
  variable hicv type integer
  variable hipdip type character length 6
  variable hiscore type float
  variable score type float
  variable conval type float
end

output
  page length 32000
  top margin 0
  left margin 0
  report to "tsw.out"
end

read into b
  unique pcl_idx
end

read into a
  wkpkg_wkpsys_idx
  wkpsys_ssn wkpsys_srf
  joining b.pcl_idx = wkpkg_proj_idx
  and wkpkg_wkpsys_idx = wkpsys_wkpkg_idx
  and wkpsys_ssn = unissn_ssn
end

read into c
  a
  priori_score
  hipri_pdip hipri_pri hipri_ok
  joining a.wkpsys_ssn = optional priori_ssn_no
  and a.wkpsys_ssn = optional hipri_ssn
end

sort by wkpkg_cmd wkpkg_cat wkpkg_no wkpsys_ssn
  hipri_ok descending hipri_pri end

format

before group of wkpkg_no
  let hiscore = -1.0
  let hissn = " "
  let hicv = 0
  let hipdip = " "

before group of wkpsys_ssn
  if wkpsys_ssn <> " " then begin
    if wkpsys_srf = 1 then let conval = 10.0
    else if wkpsys_srf = 2 then let conval = 5.0
    else if wkpsys_srf = 3 then let conval = 2.5
    else if wkpsys_srf = 4 then let conval = 1.25
    else let conval = 0.0
```

Jul 28 15:04 1986 tsw Page 2

```
let score = priori_score*conval
if score>hiscore then begin
  let hissn = wkpsys_ssn
  let hicv = wkpsys_srf
  let hiscore = score
  let hipdip = hipri_pdip
end
end

after group of wkpkg_no
print wkpkg_cmd, ":", wkpkg_cat, ":", wkpkg_subcat, ":", wkpkg_pe, ":",
wkpkg_proj, ":", wkpkg_task, ":", wkpkg_no, ":", hissn, ":", hicv, ":",
hipdip, ":", hipdip, "!\0.0!"

end
```

(Workpackages not linked to any systems, based on missing uniwkp linkages, and their funding streams. This report will be used to identify missing data elements, and to locate billpayers that might escape notice in other reports.

7/28/86. Initial report. Taken from decwbpdp. >

database manp end

define

```
param[1] pletter type character length 1
param[2] code type integer
param[3] name type character length 12
param[4] cmd type character length 12
variable pno type integer
variable pgen type integer
variable cnt type integer
variable f0 type long
variable f1 type long
variable f2 type long
variable f3 type long
variable f4 type long
variable f5 type long
variable f6 type long
variable f7 type long
variable u0 type long
variable u1 type long
variable u2 type long
variable u3 type long
variable u4 type long
variable u5 type long
variable u6 type long
variable u7 type long
```

end

input

```
prompt for pno using "Please enter the starting page number > "
end
```

output

```
left margin 0
right margin 132
report to "unlkwkp.out"
end
```

read into b

```
pcl_idx
where pcl_code = code and pcl_name = name and pcl_cmd matches cmd
end
```

read into c

```
wkpkg_no_idx wkpkg_pe wkpkg_proj wkpkg_task wkpkg_title
where uniwkp_ssn = " "
joining b.pcl_idx = wkpkg_proj_idx
and wkpkg_wkpsys_idx = optional uniwkp_wkpkg_idx
end
```

```

read into d
c
  wkpf_yr wkpf_fund wkpf_unfund
  joining c.wkpkg_no_idx = optional wkpf_wkpkg_idx
end

sort by wkpkg_pe wkpkg_proj wkpkg_task wkpkg_no wkpf_yr end

format

page header
  print column 47,"***** U N C L A S S I F I E D *****"
  skip 2 lines
  print column 45,name clipped," FUNDED WORKPACKAGES NOT LINKED TO SYSTEMS"
  print column 53,"WITH FISCAL RESOURCES SHOWN"
  skip 3 lines
  print column 90,"DEVELOPMENT SCHEDULE"
  print "-----"
  print "
  print " | PE/PROJ/TASK/WKPKG", 5 spaces, "TITLE", column 67,
  print " | FY86   FY87   FY88   FY89   FY90   FY91   FY92   FY93   |"
  print " |-----"
  print " |-----"
  print " |", column 67, "|", column 132, "|"
  let pgend = 0

page trailer
  if pgend = 0 then begin
    print " |", column 67, "|", column 132, "|"
    print "-----"
  end
  else skip 2 lines
  skip 2 lines
  print column 47,"***** U N C L A S S I F I E D *****"
  print column 100,date,2 spaces,time
  if pletter = "Z" then print column 60,pno
  else print column 60,pletter,"-",pno using "###"
  let pno = pno+1

on last record
  let pgend = 1
  print " |", column 67, "|", column 132, "|"
  print "-----"
  print " |", column 35, "Cumulative Funded Resources", column 67, "|",
  f0 using "***** ", f1 using "***** ", f2 using "***** ",
  f3 using "***** ", f4 using "***** ", f5 using "***** ",
  f6 using "***** ", f7 using "***** ", column 132, "|"
  print " |", column 35, "Unfunded Resources", column 67, "|",
  -u0 using "((((((#)", -u1 using "((((((#)", -u2 using "((((((#)",
  -u3 using "((((((#)", -u4 using "((((((#)", -u5 using "((((((#)",
  -u6 using "((((((#)", -u7 using "((((((#)", column 132, "|"
  print "-----"

```

```
before group of wkpkg_no
let cnt = cnt+1
print "!", cnt using "###. ", wkpkg_pe, 1 space, wkpkg_proj, 1 space,
    wkpkg_task clipped, 1 space, wkpkg_no, column 52, wkpkg_cmd, column 67, "!";

before group of wkpf_yr
if wkpf_yr>1985 then begin
print column (8*(wkpf_yr-1986)+68);
if wkpf_fund>0 then print wkpf_fund using "##### ";
else if wkpf_unfund>0 then print -wkpf_unfund using "((((((#)";
end
if wkpf_yr=1986 then begin
let f0 = f0+wkpf_fund
let u0 = u0+wkpf_unfund end
else if wkpf_yr=1987 then begin
let f1 = f1+wkpf_fund
let u1 = u1+wkpf_unfund end
else if wkpf_yr=1988 then begin
let f2 = f2+wkpf_fund
let u2 = u2+wkpf_unfund end
else if wkpf_yr=1989 then begin
let f3 = f3+wkpf_fund
let u3 = u3+wkpf_unfund end
else if wkpf_yr=1990 then begin
let f4 = f4+wkpf_fund
let u4 = u4+wkpf_unfund end
else if wkpf_yr=1991 then begin
let f5 = f5+wkpf_fund
let u5 = u5+wkpf_unfund end
else if wkpf_yr=1992 then begin
let f6 = f6+wkpf_fund
let u6 = u6+wkpf_unfund end
else if wkpf_yr=1993 then begin
let f7 = f7+wkpf_fund
let u7 = u7+wkpf_unfund end

after group of wkpkg_no
print column 132, "!"
print "!", 5 spaces, wkpkg_title[1, 58], column 67, "!", column 132, "!"

end
```

(This program rolls up the funding for a workpackage into a single record.)

database mamp end

define

```
param[11] base type integer
variable v      type character length 1
variable byr    type integer
variable g0     type long
variable f0     type long
variable u0     type long
variable g1     type long
variable f1     type long
variable u1     type long
variable g2     type long
variable f2     type long
variable u2     type long
variable g3     type long
variable f3     type long
variable u3     type long
variable g4     type long
variable f4     type long
variable u4     type long
variable g5     type long
variable f5     type long
variable u5     type long
variable g6     type long
variable f6     type long
variable u6     type long
variable g7     type long
variable f7     type long
variable u7     type long
```

end

output

```
top margin 0
page length 32000
left margin 0
right margin 132
report to "wkpfr.out"
```

end

read

```
wkpf_cmd wkpf_cat wkpf_wkpkg wkpf_yr wkpf_guid wkpf_fund wkpf_unfund
end
```

sort by wkpf_cmd wkpf_cat wkpf_wkpkg wkpf_yr end

format

before group of wkpf_wkpkg

```
let v = "!"
let byr = wkpf_yr
let g0 = 0
let f0 = 0
let u0 = 0
```

```
let g1 = 0
let f1 = 0
let u1 = 0
let g2 = 0
let f2 = 0
let u2 = 0
let g3 = 0
let f3 = 0
let u3 = 0
let g4 = 0
let f4 = 0
let u4 = 0
let g5 = 0
let f5 = 0
let u5 = 0
let g6 = 0
let f6 = 0
let u6 = 0
let g7 = 0
let f7 = 0
let u7 = 0
```

```
after group of wkpf_wkpkg
print wkpf_cmd, v, wkpf_cat, v, wkpf_wkpkg, v, byr, v,
g0, v, f0, v, u0, v, g1, v, f1, v, u1, v,
g2, v, f2, v, u2, v, g3, v, f3, v, u3, v,
g4, v, f4, v, u4, v, g5, v, f5, v, u5, v,
g6, v, f6, v, u6, v, g7, v, f7, v, u7, v
```

```
before group of wkpf_yr
if wkpf_yr = base then begin
let g0 = wkpf_guid
let f0 = wkpf_fund
let u0 = wkpf_unfund end
else if wkpf_yr = base+1 then begin
let g1 = wkpf_guid
let f1 = wkpf_fund
let u1 = wkpf_unfund end
else if wkpf_yr = base+2 then begin
let g2 = wkpf_guid
let f2 = wkpf_fund
let u2 = wkpf_unfund end
else if wkpf_yr = base+3 then begin
let g3 = wkpf_guid
let f3 = wkpf_fund
let u3 = wkpf_unfund end
else if wkpf_yr = base+4 then begin
let g4 = wkpf_guid
let f4 = wkpf_fund
let u4 = wkpf_unfund end
else if wkpf_yr = base+5 then begin
let g5 = wkpf_guid
let f5 = wkpf_fund
let u5 = wkpf_unfund end
else if wkpf_yr = base+6 then begin
let g6 = wkpf_guid
```

Jun 9 13:57 1986 wkpfroll Page 3

```
    let f6 = wkpf_fund
    let u6 = wkpf_unfund end
  else if wkpf_yr = base+7 then begin
    let g7 = wkpf_guid
    let f7 = wkpf_fund
    let u7 = wkpf_unfund
  end
end

end
```



```
database manp end

( modified 2/10/86 to add ma parameter, date-time stamp
  3/6/86 to add page number input
  Modified 5/23/86. changed to manp database.
  6/13/86 changed header. )

define
  variable junk      type integer
  variable pno       type integer
  param[1] pletter   type character length 1
  param[2] code       type integer
  param[3] name       type character length 12
  param[4] cmd        type character length 12
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  left margin 0
  right margin 132
  report to "wkpinde.out"
end

read into b
  proj_idx proj_title
  projctl
  where pcl_code = code and pcl_name = name and pcl_cmd matches cmd
  and (pcl_subcat = "6.3A" or pcl_subcat = "6.3B" or pcl_subcat = "6.4"
  or pcl_subcat = "6.7")
  joining pcl_idx = proj_idx
end

sort by proj_pe proj_no proj_cmd end

format

page header
  print column 47, "***** U N C L A S S I F I E D *****"
  skip 2 lines
  print 53 spaces, name clipped, " RDTE PROJECT INDEX"
  skip 3 lines
  print 20 spaces, "PE", 5 spaces, "PROJ", 23 spaces, "TITLE",
    38 spaces, "COMMAND", 8 spaces, "PAGE"
  skip 2 lines

before group of proj_no
  let junk = junk+1

before group of proj_cmd
  print 19 spaces, pcl_pe, 3 spaces, pcl_proj,
    3 spaces, proj_title, 3 spaces, pcl_cmd, 3 spaces, "----"

page trailer
```

Jun 13 17:16 1986 wkpindeX Page 2

```
skip 1 line
print column 47, "***** UNCLAS S I F I E D *****"
print column 100, date, 2 spaces, time
if pletter="Z" then print 60 spaces, pno
else print 60 spaces, pletter, "-", pno using "###"
let pno = pno+1

end
```

(7/11/86 workpackage to system cross reference. Show all workpackages in a mission area, what systems that they are linked to, what the linkage priorities of that system are, and its highest priority pdip. This will be used to determine the optimum linkage between workpackages and systems for the pdip submit process.
7/13/86 changed wkpsys_wc to wkpsys_srf.)

database mamp end

```
define
  variable pagecount type integer
  variable pno        type integer
  variable newssn     type integer
  param[1] pletter    type character length 1
  param[2] code       type integer
  param[3] name       type character length 12
  param[4] proj       type character length 4
end

input
  prompt for pno using "Please enter the starting page number > "
end

output
  right margin 132
  left margin 0
  report to "wkpsys.out"
end

read into d
  pcl_idx
  where pcl_code = code and pcl_name = name
  and pcl_cat = "6.3" and pcl_subcat <> "6.5"
  and pcl_proj matches proj
end

read into b
  proj_idx proj_title
  wkpkg_title wkpkg_task wkpkg_no_idx wkpkg_wkpsys_idx wkpkg_pdip
  joining d.pcl_idx = proj_idx
  and proj_idx = optional wkpkg_proj_idx
end

read into c
  b
  wkpsys_ssn wkpsys_srf
  uniwkp_ssn uniwkp_pdip
  joining b.wkpkg_wkpsys_idx = optional wkpsys_wkpkg_idx
  and b.wkpkg_wkpsys_idx = optional uniwkp_wkpkg_idx
end

read into a
  c
  lrrdp_title lrrdp_pdip
  lrrdppri_pri
  priori_score
```

```
      joining c.wkpsys_ssn = optional lrrdp_ssn
            and c.wkpsys_ssn = optional prior1_ssn_no
            and lrrdp_pdip = optional lrrdppri_pdip
    end

    read into e
        scl_ssn
        where scl_code = code and scl_name = name
    end

    read into c
        a
        e
        joining a.wkpsys_ssn = optional e.scl_ssn
    end

    sort by proj_cmd proj_pe proj_no wkpkg_task wkpkg_no wkpsys_ssn
        lrrdp_pdip lrrdppri_pri end

    format

    page header
        print column 47, "***** C O N F I D E N T I A L *****"
        skip 2 lines
        print column 45, "PROJECT/WORKPACKAGE/SYSTEM LINKAGE SUMMARY"
        skip 2 lines
        print 8 spaces, "COMMAND: ", proj_cmd,
            8 spaces, "PROJECT: ", proj_pe, 2 spaces, proj_no,
            10 spaces, "TITLE: ", proj_title clipped;
        if pagecount = 1 then print " (continued)" else print " "
        skip 1 line

    page trailer
        skip 2 lines
        print "Legend:", column 47, "***** C O N F I D E N T I A L *****"
        print "* - System Designated in Mission Area", column 100, date, 2 spaces, time
        print "^ - Major Designated System for Fiscal Accountability"
        print "v - Workpackage-System Rating Factor"
        if pletter="Z" then print column 60, pno
        else print column 60, pletter, "-", pno using "###"
        let pno = pno+1

    before group of proj_no
        let pagecount = 0
        skip to top of page

    before group of wkpkg_no
        skip 1 line
        let pagecount = 1
        print wkpkg_task[1,4], 1 space, wkpkg_no, 1 space, wkpkg_title, 6 spaces, "v*^",
            5 spaces, wkpkg_pdip[1,4], "-", wkpkg_pdip[5,6],
            8 spaces, "PRI", 10 spaces, "BDP"

    before group of wkpsys_ssn
        if wkpsys_ssn <> " " then begin
            print 5 spaces, wkpsys_ssn, 2 spaces, lrrdp_title, 1 space, wkpsys_srf;
```

```
    if scl_ssn<>" " then print "*"; else print " ";
    let newssn = 1
  end

before group of lrrdp_pdip
if wkpsys_ssn <> " " then begin
  if newssn = 1 then let newssn = 0 else print 81 spaces;
  if (uniwkp_ssn=wkpsys_ssn) and (uniwkp_pdip=lrrdp_pdip) then
    print "@"; else print " ";
  print 5 spaces, lrrdp_pdip[1,4], "-", lrrdp_pdip[5,6], 5 spaces, lrrdppri_pri
    using "####.##", 5 spaces, priori_score using "####.##"
  end
end

end
```

{ Work Package Appendix }

{ modified 3/6/86 to add mission area parameter and page letter parameter
and to change system data to come from ssndesc
modified 5/13/86 to run in MAMP
5/16/86 added report group (t)
6/10/86 removed report group parameter.
6/13/86 added wkpkg_class for classification. }

database mamp end

define

```
variable c1char    type character length 1
variable scnt      type integer
variable pno       type integer
param[1] pletter   type character length 1
param[2] code      type integer
param[3] name      type character length 12
param[4] cmd       type character length 12
```

end

input

```
prompt for pno using "Please enter the starting page number > "
end
```

output

```
left margin 0
right margin 132
report to "wrkapdx.out"
```

end

read into b

```
pcl_idx
where pcl_code = code and pcl_name = name and pcl_cmd matches cmd
and (pcl_subcat = "6.3A" or pcl_subcat = "6.3B" or pcl_subcat = "6.4"
or pcl_subcat = "6.7")
```

end

read into a

```
wkpkg_no_idx wkpkg_lab wkpkg_title wkpkg_pe wkpkg_proj wkpkg_task
wkpkg_wkpsys_idx wkpkg_trans_date wkpkg_class
wkpfr_f0 wkpfr_f1 wkpfr_f2 wkpfr_f3 wkpfr_f4 wkpfr_f5 wkpfr_f6 wkpfr_f7
wkpfr_u0 wkpfr_u1 wkpfr_u2 wkpfr_u3 wkpfr_u4 wkpfr_u5 wkpfr_u6 wkpfr_u7
joining b.pcl_idx = wkpkg_proj_idx
and wkpkg_no_idx = optional wkpfr_wkpkg_idx
```

end

read into e

```
wks_wkpkg_idx wks_text
where wks_yr = 1987
joining a.wkpkg_no_idx = wks_wkpkg_idx
```

end

read into b

```
a
wkd_text
```

Jun 13 17:13 1986 wrkapdx Page 2

```
e.wks_text
flagr
joining a.wkpkg_no_idx = optional wkd_idx
      and a.wkpkg_no_idx = optional e.wks_wkpkg_idx
      and a.wkpkg_no_idx = optional flagr_wkpkg_idx
end

read into c
wkpsys hipri_title ssn_da_ma
joining b.wkpkg_wkpsys_idx = wkpsys_wkpkg_idx
      and wkpsys_ssn = ssn_ssn
      and wkpsys_ssn = hipri_ssn
end

read into a
b
c.wkpsys_ssn c.hipri_title c.ssn_da_ma
joining b.wkpkg_wkpsys_idx = optional c.wkpsys_wkpkg_idx
end

sort by wkpkg_pe wkpkg_proj wkpkg_cmd wkpkg_task wkpkg_no wkpsys_ssn end

format

page header
let clchar = " "
if wkpkg_class="S" then begin
let clchar = "S"
print column 52,"***** S E C R E T *****"
end
else if wkpkg_class="C" then begin
let clchar = "C"
print column 47,"***** C O N F I D E N T I A L *****"
end
else print column 47,"***** U N C L A S S I F I E D *****"
skip 2 lines
print column 45,"WORKPACKAGE SUMMARY DATA FOR ",wkpkg_cmd clipped;
if (wkpkg_lab<>" " and wkpkg_lab<>wkpkg_cmd) then print " / ",wkpkg_lab
else print ""
skip 2 lines

page trailer
if clchar = "S" then print column 52,"***** S E C R E T *****"
else if clchar = "C" then print column 47,"***** C O N F I D E N T I A L *****"
else print column 47,"***** U N C L A S S I F I E D *****"
print column 100,date,2 spaces,time
if pletter="Z" then print column 60,pno
else print column 60,pletter,"-",pno using "###"
let pno = pno+1

before group of wkpkg_no
skip to top of page
let scnt = 0
print "PE/Project/Task: ",wkpkg_pe,1 space,wkpkg_proj,1 space,
      wkpkg_task clipped," : WP ",wkpkg_no,column 67,"Title: ",wkpkg_title
print "Funded by: ",name;
```

```

print column 30, "Transition Date: ", wkpkg_trans_date
skip 3 lines
print "          FY86  FY87  FY88  FY89  FY90  FY91  FY92  FY93";
print column 90, "Description"
print "          -----"
print "Funded:   ", wkpfr_f0 using "*****",
              wkpfr_f1 using "*****",
              wkpfr_f2 using "*****",
              wkpfr_f3 using "*****",
              wkpfr_f4 using "*****",
              wkpfr_f5 using "*****",
              wkpfr_f6 using "*****",
              wkpfr_f7 using "*****";

print column 72, wkd_text[1, 60]
print "Unfunded: ", wkpfr_u0 using "*****",
              wkpfr_u1 using "*****",
              wkpfr_u2 using "*****",
              wkpfr_u3 using "*****",
              wkpfr_u4 using "*****",
              wkpfr_u5 using "*****",
              wkpfr_u6 using "*****",
              wkpfr_u7 using "*****";

print column 72, wkd_text[61, 120]
print column 72, wkd_text[121, 180]
print "Flags",
print column 72, wkd_text[181, 240]
if flagr1_n<>" " then begin
  print flagr1_n, 5 spaces;
  print flagr1_0 using "*****", flagr1_1 using "*****",
    flagr1_2 using "*****", flagr1_3 using "*****",
    flagr1_4 using "*****", flagr1_5 using "*****",
    flagr1_6 using "*****", flagr1_7 using "*****";
end
print column 72, wkd_text[241, 300]
if flagr2_n<>" " then begin
  print flagr2_n, 5 spaces;
  print flagr2_0 using "*****", flagr2_1 using "*****",
    flagr2_2 using "*****", flagr2_3 using "*****",
    flagr2_4 using "*****", flagr2_5 using "*****",
    flagr2_6 using "*****", flagr2_7 using "*****";
end
print column 72, wkd_text[301, 360]
if flagr3_n<>" " then begin
  print flagr3_n, 5 spaces;
  print flagr3_0 using "*****", flagr3_1 using "*****",
    flagr3_2 using "*****", flagr3_3 using "*****",
    flagr3_4 using "*****", flagr3_5 using "*****",
    flagr3_6 using "*****", flagr3_7 using "*****";
end
print column 72, wkd_text[361, 420]
if flagr4_n<>" " then begin
  print flagr4_n, 5 spaces;
  print flagr4_0 using "*****", flagr4_1 using "*****",
    flagr4_2 using "*****", flagr4_3 using "*****",
    flagr4_4 using "*****", flagr4_5 using "*****",
    flagr4_6 using "*****", flagr4_7 using "*****";
end

```



```

end
print column 72, wkd_text[421,480]
if flagr5_n<>" " then begin
  print flagr5_n, 5 spaces;
  print flagr5_0 using "#####", flagr5_1 using "#####",
    flagr5_2 using "#####", flagr5_3 using "#####",
    flagr5_4 using "#####", flagr5_5 using "#####",
    flagr5_6 using "#####", flagr5_7 using "#####";
end
print column 72, wkd_text[481,540]
if flagr6_n<>" " then begin
  print flagr6_n, 5 spaces;
  print flagr6_0 using "#####", flagr6_1 using "#####",
    flagr6_2 using "#####", flagr6_3 using "#####",
    flagr6_4 using "#####", flagr6_5 using "#####",
    flagr6_6 using "#####", flagr6_7 using "#####";
end
print column 72, wkd_text[541,600]
skip 3 lines
print column 10, "Systems Supported (SSN/TITLE/DA MA)";
print column 90, "1987 Workplan"
skip 1 line

before group of wkpsys_ssn
if scnt:15 then print wkpsys_ssn, 2 spaces, hipri_title[1, 50], 1 space,
  ssn_da_ma, column 72;
let scnt = scnt+1
if scnt:11 then begin
  if scnt = 1 then print wks_text[1, 60]
  else if scnt = 2 then print wks_text[61, 120]
  else if scnt = 3 then print wks_text[121, 180]
  else if scnt = 4 then print wks_text[181, 240]
  else if scnt = 5 then print wks_text[241, 300]
  else if scnt = 6 then print wks_text[301, 360]
  else if scnt = 7 then print wks_text[361, 420]
  else if scnt = 8 then print wks_text[421, 480]
  else if scnt = 9 then print wks_text[481, 540]
  else if scnt = 10 then print wks_text[541, 600]
end
else print ""

after group of wkpkg_no
if scnt=0 then begin
  print column 72, wks_text[1, 60]
  print column 72, wks_text[61, 120]
  print column 72, wks_text[121, 180]
  print column 72, wks_text[181, 240]
  print column 72, wks_text[241, 300]
  print column 72, wks_text[301, 360]
  print column 72, wks_text[361, 420]
  print column 72, wks_text[421, 480]
  print column 72, wks_text[481, 540]
  print column 72, wks_text[541, 600]
end
else if scnt=1 then begin
  print column 72, wks_text[61, 120]

```

```
print column 72.wks_text[121,180]
print column 72.wks_text[181,240]
print column 72.wks_text[241,300]
print column 72.wks_text[301,360]
print column 72.wks_text[361,420]
print column 72.wks_text[421,480]
print column 72.wks_text[481,540]
print column 72.wks_text[541,600]
end
else if scnt=2 then begin
print column 72.wks_text[121,180]
print column 72.wks_text[181,240]
print column 72.wks_text[241,300]
print column 72.wks_text[301,360]
print column 72.wks_text[361,420]
print column 72.wks_text[421,480]
print column 72.wks_text[481,540]
print column 72.wks_text[541,600]
end
else if scnt=3 then begin
print column 72.wks_text[181,240]
print column 72.wks_text[241,300]
print column 72.wks_text[301,360]
print column 72.wks_text[361,420]
print column 72.wks_text[421,480]
print column 72.wks_text[481,540]
print column 72.wks_text[541,600]
end
else if scnt=4 then begin
print column 72.wks_text[241,300]
print column 72.wks_text[301,360]
print column 72.wks_text[361,420]
print column 72.wks_text[421,480]
print column 72.wks_text[481,540]
print column 72.wks_text[541,600]
end
else if scnt=5 then begin
print column 72.wks_text[301,360]
print column 72.wks_text[361,420]
print column 72.wks_text[421,480]
print column 72.wks_text[481,540]
print column 72.wks_text[541,600]
end
else if scnt=6 then begin
print column 72.wks_text[361,420]
print column 72.wks_text[421,480]
print column 72.wks_text[481,540]
print column 72.wks_text[541,600]
end
else if scnt=7 then begin
print column 72.wks_text[421,480]
print column 72.wks_text[481,540]
print column 72.wks_text[541,600]
end
else if scnt=8 then begin
print column 72.wks_text[481,540]
```

Jun 13 17:13 1986 wrkapdx Page 6

```
    print column 72.wks_text[541,600]
end
else if scnt=9 then begin
    print column 72.wks_text[541,600]
end
end
```

{5/16/1986: This report generates input for the flagr database file
which is used in the wrkapdx report.}

database manp end

define

```
param[1] base type integer
variable v type character length 1
variable cnt type integer
variable f0 type long
variable f1 type long
variable f2 type long
variable f3 type long
variable f4 type long
variable f5 type long
variable f6 type long
variable f7 type long
```

end

output

```
top margin 0
page length 32000
left margin 0
report to "xferflag.out"
```

end

read into b

```
unique pcl_idx
```

end

read into a

```
flag_cmd flag_cat flag_wkpkg flag_name flag_yr flag_fund
joining b.pcl_idx = wkpkg_proj_idx
and wkpkg_no_idx = flag_wkpkg_idx
```

end

sort by flag_cmd flag_cat flag_wkpkg flag_name flag_yr end

format

before group of flag_wkpkg

```
let cnt=0
let v = "!"
print flag_cmd,v,flag_cat,v,flag_wkpkg,v;
```

before group of flag_name

```
let f0 = 0
let f1 = 0
let f2 = 0
let f3 = 0
let f4 = 0
let f5 = 0
let f6 = 0
let f7 = 0
```

after group of flag_name

```
    let cnt = cnt+1
    if cnt<7 then print flag_name, v, f0, v, f1, v, f2, v, f3, v, f4, v, f5, v, f6, v, f7, v;

after group of flag_wkpkg
while cnt<6 do begin
    print "    :0:0:0:0:0:0:0:0:0:0:";
    let cnt=cnt+1
end
print ""

before group of flag_yr
if flag_yr = base then let f0 = flag_fund
else if flag_yr = base+1 then let f1 = flag_fund
else if flag_yr = base+2 then let f2 = flag_fund
else if flag_yr = base+3 then let f3 = flag_fund
else if flag_yr = base+4 then let f4 = flag_fund
else if flag_yr = base+5 then let f5 = flag_fund
else if flag_yr = base+6 then let f6 = flag_fund
else if flag_yr = base+7 then let f7 = flag_fund

end
```

DATE
FILMED
= 8